



TUGAS AKHIR - TE 145561

**RANCANG BANGUN DETEKTOR ARITHMIA
MENGUNAKAN QRS *DETECTOR* BERBASIS
MIKROKONTROLER**

Abdurrachman Hakim
NRP 2213039044

Dosen Pembimbing
Rachmad Setiawan, ST., MT.
Eko Pujiyatno Matni, S.Pd.

PROGRAM STUDI ELEKTRONIKA INDUSTRI
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 145561

***ARRHYTHMIA DETECTOR BUILDING DESIGN USING
QRS DETECTOR BASED ON MICROCONTROLLER***

Abdurrachman Hakim
NRP 2213039044

Supervisor
Rachmad Setiawan, ST., MT.
Eko Pujiyatno Matni, S.Pd.

***INDUSTRIAL ELECTRONICS STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2017***

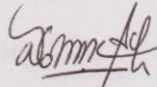
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Rancang Bangun Detektor Arithmia Menggunakan QRS Detector Berbasis Mikrokontroler**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 17 Juli 2017



Abdurrachman Hakim
NRP 2213039044

-----Halaman ini sengaja dikosongkan-----

**RANCANG BANGUN DETEKTOR ARITHMIA
MENGUNAKAN QRS DETECTOR BERBASIS
MIKROKONTROLER**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Program Studi Elektronika Industri
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II

Rachmad Setiawan, ST., MT.
NIP. 1969 05 29 1995 12 1001

Eko Pujiyatno Matni, S.Pd.
NIP. 1971 03 30 1994 03 1002

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN DETEKTOR ARITHMIA MENGUNAKAN QRS *DETECTOR* BERBASIS MIKROKONTROLER

Nama Mahasiswa : Abdurrachman Hakim
Dosen Pembimbing 1 : Rachmad Setiawan, ST., MT
Dosen Pembimbing 2 : Eko Pujiyatno Matni, S.Pd

ABSTRAK

Penelitian ini membahas mengenai detektor arithmia yang merupakan salah satu cara mendeteksi kelainan jantung. Dalam suatu proses untuk mengetahui arithmia atau kelainan jantung, ada pengukuran beberapa kali untuk mengetahui hasil yang maksimal dan terkadang terdapat interferensi sinyal pergerakan. Komponen utama yang terdapat pada sebuah sinyal ECG adalah kompleks QRS. Faktor penting pada diagnosa ini adalah pengukuran *interval* waktu antara puncak-puncak R kompleks QRS. Alat ini menggunakan modul ECG (AD8232) yang dihubungkan dengan mikrokontroler, data dikirim secara serial dan ditampilkan di komputer. Alat ini dengan menggunakan mikrokontroler Arduino UNO, *software* Delphi untuk mendapatkan kompleks QRSnya sebagai *software QRS Detector*. Cara untuk mengatasi kendala tersebut yaitu dengan menggunakan metode *QRS detector* yaitu dengan memasukkan persamaan atau fungsi *low pass filter*, *high pass filter*, *derivative*, *squaring* dan *moving window* yang tercantum pada buku *biomedical signal processing* karya Wille Tompkins. Hasil yang ditampilkan berupa sinyal ECG yang telah di *filter* melalui persamaan *low pass filter* dan *high pass filter*. Kemudian didiferensiasi menghasilkan sinyal *derivative*. Setelah itu di kuadratkan menghasilkan sinyal *squaring* dan dimasukkan persamaan *moving window* untuk menghasilkan sinyal *moving window*. Dari keseluruhan proses sinyal tersebut didapatkan nilai jarak R–R yaitu berkisar 180 Hz. Sedangkan data yang didapatkan setiap detiknya yaitu 200. Sehingga per menitnya mendapatkan 12000 Hz. Maka hasil *heart rate* nya yaitu 12000 dibagi dengan 180 yaitu berkisar 60–70 Bpm.

Kata Kunci : Sinyal ECG, Mikrokontroler, QRS, Arithmia, *Software* Delphi

-----Halaman ini sengaja dikosongkan-----

ARRHYTHMIA DETECTOR BUILDING DESIGN USING QRS DETECTOR BASED ON MICROCONTROLLER

Name : Abdurrachman Hakim
Supervisor 1 : Rachmad Setiawan, ST., MT
Supervisor 2 : Eko Pujiyatno Matni, S.Pd

ABSTRACT

This research discusses the detector arrhythmia which is one way to detect heart abnormalities. In a process to determine arrhythmia or heart abnormality, there are measurements several times to find out the maximum results and sometimes there is movement signal interference. The main component of an ECG signal is the QRS complex. An important factor in this diagnosis is the measurement of the time interval between the peaks of the QRS complex. This tool uses ECG module (AD8232) connected with microcontroller, data is sent serially and displayed on computer. This tool uses Arduino UNO microcontroller, Delphi software to get its QRS complex as QRS Detector software. The QRS Detector method is by entering the equation or low pass filter function, high pass filter, derivative, squaring and moving window which are listed in Wille Tompkins' biomedical signal processing book. Results shown are ECG signals that have been filtered through low pass filter equations and high pass filters. Then in differentiation it produces derivative signals. After that in squared generate squaring signal and inserted moving window equations to produce moving window signals. From the whole process of the signal got the distance of R-R is 180. While the data obtained every second is 200. So per minute get 12000 Hz. So the result of his heart rate is 12000 divided by 180 ranging from 60 to 70 Bpm

Keywords : *ECG Signals, Microcontroller, QRS, Arrhythmia, Delphi Software*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Nabi Besar Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Elektronika Industri, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN DETEKTOR ARITHMIA MENGUNAKAN QRS *DETECTOR* BERBASIS MIKROKONTROLER

Dengan terselesaikannya Tugas Akhir ini, Penulis menyampaikan terima kasih yang sebesar-besarnya kepada Bapak dan ibu saya yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti. Bapak Rachmad Setiawan, ST., MT. atas segala bimbingan ilmu, moral dari awal hingga terselesaikannya tugas akhir ini. Bapak Eko Pujiyatno Matni, S.Pd. selaku dosen pembimbing dari BLKIP. Bapak Ir. Josaphat Pramudijanto, M.Eng., Ir. Djoko Suprajitno Rahardjo, MT., Yunafi'atul Aniroh, ST., M.Sc. selaku dosen penguji. Habibi Idrus bin Muhammad Alaydrus yang memberikan doa dan semangat kepada penulis. Adinda Zuhrofatul Ulwiyah yang selalu memberikan doa, semangat, dan dukungannya kepada penulis. Teman – teman UKM Cinta Rebana ITS yang selalu memberi siraman spiritual dan semangat kepada penulis. Komunitas Koloni Kecoa yang selalu siap menjadi teman diskusi hingga malam hari. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	iii
PERNYATAAN KEASILAN TUGAS AKHIR.....	Error! Bookmark not defined.
HALAMAN PENGESAHAN.....	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Laporan	3
1.7 Relevansi	4
BAB II TEORI DASAR	5
2.1 Mikrokontroler Arduino UNO	5
2.2 QRS Detector	7
2.2.1 Low Pass Filter (LPF).....	9
2.2.2 High Pass Filter (HPF)	10
2.2.3 Derivative	10
2.2.4 Squaring	11
2.2.5 Moving Window.....	12
2.3 Elektrokardiograf (ECG).....	13
2.3.1 Modul ECG AD-8232	14
2.3.2 Sensor Elektroda	15
2.3.3 Monitoring Sinyal ECG.....	15
2.3.4 Gelombang Elektrokardiogram (ECG).....	17
2.4 USB UART FT232RL	18
2.5 Rangkaian Low Pass Filter	20

2.6	<i>Software Delphi</i>	22
2.7	Jantung	26
2.8	Arithmia	29
BAB III PERANCANGAN SISTEM.....		33
3.1	Perancangan Elektronik	33
3.1.1	<i>Setting Point</i> Mikrokontroler	34
3.1.2	Rangkaian <i>Low Pass Filter</i>	35
3.1.3	Perancangan Modul AD8232 Dihubungkan Rangkaian LPF	36
3.1.4	Perancangan Rangkaian LPF Dihubungkan Mikrokontroler	36
3.1.5	Perancangan Modul AD8232 Dihubungkan Mikrokontroler	37
3.1.6	Perancangan FT232RL Dihubungkan Arduino UNO	38
3.2	Perancangan <i>Software</i>	39
3.2.1	Program Pembacaan Modul AD8232 Mikrokontroler	40
3.2.2	Program Pembacaan Nilai Data <i>Integer</i> pada Mikrokontroler	42
3.2.3	Perancangan <i>Software QRS Detector</i>	44
BAB IV PENGUKURAN DAN ANALISA		49
4.1	Pengujian Sensor Elektroda dan <i>Board</i> AD8232	49
4.1.1	Hasil Uji Elektroda dan AD8232 Cara 1	50
4.1.2	Hasil Uji Elektroda dan AD8232 Cara 2	50
4.2	Pengujian Hasil AD8232 yang Dihubungkan Rangkaian <i>Low Pass Filter</i>	51
4.2.1	Hasil Uji Sinyal ECG AD8232 Dihubungkan Rangkaian <i>Low Pass Filter</i> Cara 1	51
4.2.2	Hasil Uji Sinyal ECG AD8232 Dihubungkan Rangkaian <i>Low Pass Filter</i> Cara 2	52
4.3	Pengujian Rangkaian <i>Low Pass Filter</i>	52
4.4	Pengujian Pengiriman Data Oleh USB FT232RL	53
4.5	Pengujian Pembacaan Data <i>Integer</i> Arduino yang Dihubungkan Modul AD8232	55
4.6	Pengujian Instrument Detektor Arithmia dengan <i>Software QRS Detector</i>	56
4.6.1	Pembacaan <i>Setting Point</i>	56
4.6.2	Pembacaan Tampilan Respon ECG	56
4.7	<i>Signal Processing</i>	57

4.7.1 Hasil Sinyal ECG	58
4.7.2 Hasil Sinyal LPF ECG	58
4.7.3 Hasil Sinyal HPF ECG	59
4.7.4 Hasil Sinyal <i>Derivative</i>	60
4.7.5 Hasil Sinyal <i>Squaring</i>	60
4.7.6 Hasil Sinyal <i>Moving</i>	61
BAB V PENUTUP.....	63
5.1 Kesimpulan	63
5.2 Saran.....	63
DAFTAR PUSTAKA	65
LAMPIRAN A <i>DATASHEET</i>	A-1
LAMPIRAN B PROGRAM	B-1
LAMPIRAN C GAMBAR.....	C-1
DAFTAR RIWAYAT HIDUP	D-1

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1	Arduino UNO7
Gambar 2.2	Tahap Penyaringan QRS <i>Detector</i>8
Gambar 2.3	Contoh Elektrokardiograf dengan Sampel 200 Per Detik9
Gambar 2.4	Sinyal <i>Low Pass Filter</i> ECG9
Gambar 2.5	Sinyal <i>High Pass Filter</i> ECG10
Gambar 2.6	Sinyal <i>Derivative</i>11
Gambar 2.7	Sinyal <i>Squaring</i>11
Gambar 2.8	Sinyal <i>Moving Window</i>12
Gambar 2.9	Hubungan Kompleks QRS dengan Bentuk Gelombang Integrasi Bergerak13
Gambar 2.10	AD8232 <i>Heart Rate Monitor</i>14
Gambar 2.11	Sensor Elektroda15
Gambar 2.12	Konfigurasi Pemasangan Elektroda16
Gambar 2.13	Gelombang <i>Output</i> ECG18
Gambar 2.14	Gelombang Sinyal ECG18
Gambar 2.15	Konfigurasi Pin FT232RL19
Gambar 2.16	<i>Board</i> FT323RL20
Gambar 2.17	Rangkaian Dasar dan Respon Frekuensi <i>Low Pass Filter</i> RC21
Gambar 2.18	Jendela <i>Software</i> Delphi23
Gambar 2.19	<i>Menu</i> Delphi23
Gambar 2.20	<i>Toolbar</i> Delphi23
Gambar 2.21	<i>Component</i> Delphi24
Gambar 2.22	<i>Object Tree View</i> Delphi24
Gambar 2.23	<i>Object Inspector</i> Delphi24
Gambar 2.24	<i>Form Designer</i> Delphi25
Gambar 2.25	<i>Code Explorer</i> Delphi25
Gambar 2.26	Gambar <i>Code Editor</i> Delphi26
Gambar 2.27	Struktur Jantung29
Gambar 3.1	Pin ATmega 32835
Gambar 3.2	Rangkaian <i>Low Pass Filter</i>35
Gambar 3.3	Konfigurasi AD8232 ke Rangkaian LPF36

Gambar 3.4	Konfigurasi Rangkaian LPF ke Mikrokontroler	37
Gambar 3.5	Konfigurasi Arduino ke AD8232	38
Gambar 3.6	Konfigurasi Arduino ke USB TTL FT232RL	39
Gambar 3.7	Program Arduino untuk Menampilkan Sinyal ECG AD8232 pada Serial <i>Plotter</i>	40
Gambar 3.8	Lanjutan Program Arduino untuk Menampilkan Sinyal ECG AD8232 pada Serial <i>Plotter</i>	41
Gambar 3.9	Perintah Membaca <i>Output</i> Analog pada AD8232	41
Gambar 3.10	Program Pembacaan Nilai Data <i>Integer</i>	42
Gambar 3.11	Lanjutan Program Pembacaan Nilai Data <i>Integer</i>	43
Gambar 3.12	Perintah Pembacaan Nilai Data <i>Integer</i>	43
Gambar 3.13	Program Pembacaan Nilai <i>Setting Point</i>	44
Gambar 3.14	Program Pembacaan Tampilan ECG 400 Data.....	45
Gambar 3.15	Program Pembacaan Sinyal ECG	45
Gambar 3.16	Program Pembacaan Sinyal HPF.....	45
Gambar 3.17	Program Pembacaan Sinyal LPF	46
Gambar 3.18	Program Pembacaan Sinyal <i>Derivative</i>	46
Gambar 3.19	Pogram Pembacaan Sinyal <i>Squaring</i>	47
Gambar 3.20	Program Pembacaan Sinyal <i>Moving Window</i>	47
Gambar 3.21	Tampilan <i>Software QRS Detector</i> pada Delphi.....	48
Gambar 4.1	Program Pengujian Elektroda dan AD8232.....	49
Gambar 4.2	Hasil Uji Elektroda Cara 1.....	50
Gambar 4.3	Hasil Uji Elektroda Cara 2.....	50
Gambar 4.4	Hasil Uji AD8232 Dihubungkan Rangkaian LPF Cara 1	51
Gambar 4.5	Hasil Uji AD8232 Dihubungkan Rangkaian LPF Cara 2	52
Gambar 4.6	Hasil Tampilan ECG Sebelum Dihubungkan Rangkaian <i>Low Pass Filter</i>	53
Gambar 4.7	Hasil Tampilan ECG Sesudah Dihubungkan Rangkaian <i>Low Pass Filter</i>	53
Gambar 4.8	Hasil Uji USB FT232 RL	54
Gambar 4.9	Program Pengujian USB.....	54
Gambar 4.10	Lanjutan Program Pengujian USB.....	55
Gambar 4.11	Hasil Pembacaan Data <i>Integer</i>	55
Gambar 4.12	Hasil Tampilan Pembacaan <i>Setting Point</i>	56
Gambar 4.13	Hasil Tampilan Respon ECG.....	57
Gambar 4.14	Tampilan Sinyal ECG.....	58
Gambar 4.15	Tampilan Sinyal LPF ECG	59

Gambar 4.16	Tampilan Sinyal HPF ECG	59
Gambar 4.17	Tampilan Sinyal <i>Derivative</i>	60
Gambar 4.18	Tampilan Sinyal <i>Squaring</i>	61
Gambar 4.19	Tampilan Sinyal <i>Moving</i>	61

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

HALAMAN

Tabel 2.1	Spesifikasi Arduino.....	5
Tabel 2.2	Pin AD8232	15
Tabel 3.1	Konfigurasi <i>Port</i> Mikrokontroler.....	34
Tabel 3.2	Konfigurasi Arduino ke AD8232.....	37
Tabel 3.3	Konfigurasi Pin USB FT232RL ke Pin Arduino UNO.....	39

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini dalam bidang kedokteran banyak sekali digunakan alat biomedik sebagai alat bantu dalam proses pemeriksaan. Salah satu contohnya adalah untuk mendeteksi kelainan jantung. Dalam suatu proses untuk mengetahui arithmia atau kelainan jantung, ada pengukuran beberapa kali untuk mengetahui hasil yang maksimal. Tetapi masih ada kendala dalam proses pengukurannya. Misalnya, ketika pengukuran terdapat sinyal pergerakan dari pasien, maka dibuat instrumentasi khusus untuk ECG (Elektrokardiograf) disertai dengan frekuensi sampling dengan besaran tertentu. Oleh karena itu diperlukan suatu alat yang dapat menjadi solusi untuk permasalahan tersebut.

QRS *detector* merupakan metode yang umum digunakan untuk mendiagnosa kondisi jantung seorang pasien. Komponen utama yang terdapat pada sebuah sinyal ECG adalah kompleks QRS yang menggambarkan proses depolarisasi pada ventrikel. Faktor penting pada diagnosa ini adalah pengukuran *interval* waktu antara puncak-puncak R kompleks QRS. Pada saat pengukuran, biasanya terdapat pergerakan dari pasien, maka dibuatlah instrumentasi khusus untuk ECG disertai frekuensi sampling dengan besaran tertentu, biasanya 200 Hz. Oleh karena itu saya ingin membuktikan ide ini dengan membuat “Rancang Bangun Detektor Arithmia menggunakan QRS *Detector* berbasis Mikrokontroler” yaitu dengan menggunakan mikrokontroler, *software* Delphi untuk mendapatkan kompleks QRS, menghitung arithmia dan *database* data pasien.

Alat ini menggunakan modul ECG (AD8232) yang dihubungkan dengan mikrokontroler, data dikirim secara serial dan ditampilkan di komputer. Diharapkan dengan alat tersebut, dapat memberikan kemudahan kepada pengukur untuk mengetahui kelainan jantung.

1.2 Rumusan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini adalah instrument untuk mendapatkan sinyal ECG jantung, cara mengatasi kendala – kendala dikarenakan sinyal otot lainnya dan menghitung hasil *heart rate* yang dihasilkan oleh jarak puncak R–R setiap menit.

1.3 Batasan Masalah

Alat ini menggunakan modul ECG (AD8232) yang dihubungkan dengan mikrokontroler menggunakan USB TTL kemudian ditampilkan ke PC menggunakan *software* Delphi.

1.4 Tujuan

Tujuan kami menulis Tugas Akhir ini adalah sebagai berikut :

- Merancang dan membuat “Rancang Bangun Detektor Arithmia menggunakan QRS *Detector* berbasis Mikrokontroler”.
- Memberi kemudahan kepada pihak pengukur untuk mengetahui arithmia.

1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu studi literatur, pemodelan sistem, perancangan algoritma kontrol, simulasi hasil desain, implementasi dan analisis data, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

- **Studi Pustaka dan Survey Data Awal**

Studi pustaka yang dilakukan dalam hal ini adalah dengan mencari literatur di ruang baca, perpustakaan atau di internet mengenai mikrokontroler, arithmia, QRS *detector* dan *database*. Untuk survey data awal yang dilakukan dalam hal ini adalah dengan melakukan survey mengenai proses pengukuran dan metode yang digunakan pada QRS *detector*.

- **Perencanaan dan Pembuatan Alat**

Tahapan ini dilakukan setelah mendapat informasi dari referensi diatas. Perencanaan alat meliputi perancangan instrumentasi khusus yaitu “Rancang Bangun Detektor Arithmia menggunakan QRS *Detector* dengan Mikrokontroler“ yaitu modul ECG (AD8232) yang di hubungkan dengan *low pass filter* yang dibuat kemudian di hubungkan dengan mikrokontroler arduino.

- **Perencanaan dan Pembuatan *Software***

Pembuatan *software* dilakukan setelah perancangan alat selesai dilakukan, pembuatan *software* ini dilakukan pada mikrokontroler menggunakan *software* IDE dan Delphi.

- **Pengukuran dan Analisa**

Uji coba dan analisa dilakukan dengan cara mengaplikasikan secara langsung instrumentasi khusus yang dibuat yaitu “Rancang bangun Detektor Arithmia menggunakan QRS *Detector* berbasis Mikrokontroler”.

- **Penyusunan Laporan**

Penyusunan laporan dilakukan setelah beberapa pengujian instrumentasi rancang bangun detektor arithmia yang didapat sudah mencukupi.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, mikrokontroler arduino, elektrokardiograf, USB FT232RL, rangkaian *low pass filter*, dan *software* Delphi

Bab III Perancangan dan Pembuatan Alat

Bab ini membahas perancangan elektronik (*hardware*) dan *software* untuk QRS *Detector* berdasarkan teori dasar pada Bab II.

Bab IV Pengukuran dan Analisa

Bab ini memuat hasil pengujian dan implementasi serta analisis dari hasil tersebut.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan menjadi referensi alat bantu dalam proses pengukuran untuk mengetahui arithmia, sebagai cara untuk menghindari pendapatan hasil pengukuran yang kurang maksimal, dan Untuk memperkecil kemungkinan kesalahan pengukuran.

BAB II

TEORI DASAR

Pada bab ini dibahas mengenai teori - teori yang menunjang dalam proses pembuatan alat Tugas Akhir. Teori yang menunjang dalam Tugas Akhir ini diantaranya adalah mikrokontroler Arduino UNO, QRS *detector*, elektrokardiograf (ECG) meliputi modul ECG AD8232, elektroda, monitoring sinyal ECG, gelombang ECG, USB FT232RL, rangkaian *low pass filter*, *software* Delphi, jantung dan arithmia.

2.1 Mikrokontroler Arduino UNO [1] [2]

Mikrokontroler adalah sebuah *chip* yang berfungsi sebagai pengontrol rangkaian elektronik dan umunya dapat menyimpan program didalamnya. Mikrokontroler umumnya terdiri dari CPU (*Central Processing Unit*), memori, I/O tertentu dan unit pendukung seperti *Analog to Digital Converter* (ADC) yang sudah terintegrasi di dalamnya. Program disimpan dalam bentuk RAM, *Nor Flash*, OTP ROM yang disertakan dalam *chip*.

Salah satunya adalah mikrokontroler Arduino UNO yang masih dari tipe mikrokontroler ATMega yang berbasis *chip* ATMega 328P. Spesifikasi dari Arduino UNO seperti Tabel 2.1.

Tabel 2.1 Spesifikasi Arduino

Mikrokontroler	ATMega328
Operasi tegangan	5 V
<i>Input</i> tegangan	Disarankan 7-11 V
<i>Input</i> tegangan batas	6-20 V
Pin I/O digital	14 (6 bisa untuk PWM)
Pin Analog	6
Arus DC tiap pin I/O	50 mA
Arus DC ketika 3.3V	50 mA
Memori <i>flash</i>	32 KB (ATMega 328) dan 0,5 KB digunakan oleh <i>bootloader</i>
SRAM	2 KB (ATMega 328)
EEPROM	1 KB (ATMega 328)
Kecepatan <i>clock</i>	16 MHz
Dimensi	68,6 mm x 53,4 mm
Berat	25 Kg

Berdasarkan Gambar 2.1, UNO memiliki 14 pin digital *input / output* (dimana 6 dapat digunakan sebagai *output* PWM), 6 *input* analog, resonator keramik 16 MHz, koneksi USB, jack listrik, *header* ICSP, dan tombol *reset*. Arduino UNO dibangun berdasarkan apa yang diperlukan untuk mendukung mikrokontroler, sumber daya bisa menggunakan *power* USB (jika terhubung ke komputer dengan kabel USB) dan juga dengan adaptor atau baterai. Arduino UNO juga dapat diaktifkan melalui koneksi USB atau dengan catu daya *eksternal*. Sumber daya dipilih secara otomatis. Untuk sumber daya *eksternal* (non USB) dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor ini dapat dihubungkan dengan memasukkan 2.1 mm jack DC ke colokan listrik *board*. Baterai dapat dimasukkan pada pin header GND dan Vin dari konektor daya. *Board* dapat beroperasi pada pasokan *eksternal* dari 6 V sampai 20 V. Jika menggunakan tegangan kurang dari 6 V mungkin tidak stabil. Jika menggunakan lebih dari 12 V, regulator tegangan bisa panas dan merusak papan. Rentang yang dianjurkan adalah 7 V sampai 12 V.

Masing-masing dari 14 pin digital UNO dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Mereka beroperasi pada tegangan 5 V. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up internal* (terputus secara *default*) dari 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi spesial:

- Serial: pin 0 (RX) dan 1 (TX) digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL.
- *Eksternal* Interupsi: Pin 2 dan 3 dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah (*low value*), *rising* atau *falling edge*, atau perubahan nilai.
- PWM: Pin 3, 5, 6, 9, 10, dan 11 menyediakan 8-bit PWM dengan fungsi *analogWrite()*
- SPI: pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) mendukung komunikasi SPI dengan menggunakan perpustakaan SPI
- LED: pin 13. Built-in LED terhubung ke pin digital 13. LED menyala ketika diberi nilai *high*

Arduino UNO memiliki 6 *input* analog, berlabel A0 sampai A5, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara *default* mereka mengukur dari *ground* sampai 5 V, perubahan tegangan maksimal menggunakan pin AREF. Selain itu,

beberapa pin tersebut memiliki spesialisasi fungsi, yaitu TWI: pin A4 atau SDA dan A5 atau SCL mendukung komunikasi TWI menggunakan perpustakaan *Wire*.

Ada beberapa pin lainnya yang tertulis di *board*:

- **AREF.** Tegangan referensi untuk *input* analog. Dapat digunakan dengan fungsi *analogReference()*.
- **Reset.** Gunakan LOW untuk melakukan *reset* mikrokontroler. Biasanya digunakan untuk menambahkan tombol *reset*.

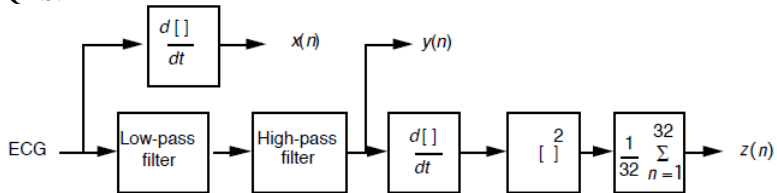


Gambar 2.1 Arduino UNO

2.2 QRS Detector [3]

Suatu detektor QRS umumnya terdapat tahapan-tahapan pemrosesan berikut, yaitu tahap pemrosesan awal (*preprocessing stage*) atau tahap untuk mengekstrak kompleks QRS sinyal ECG. Tahap ini terdiri atas proses pemfilteran linier dan nonlinier. Tahap kedua adalah tahap keputusan (*decision stage*) atau algoritma aturan keputusan yang digunakan untuk mendeteksi gelombang R. Berdasarkan Gambar 2.2, algoritma detektor kompleks QRS menggunakan *filter* untuk menghilangkan komponen-komponen gelombang lain dan artefak pada frekuensi puncak tersebut seperti gelombang P, gelombang T, penyimpangan garis dasar (*baseline drift*) dan *noise*. Untuk tujuan tersebut biasanya digunakan *filter* lolos tinggi (*highpass filter*) untuk menghilangkan atau melemahkan gelombang P, T dan memperbaiki penyimpangan garis dasar serta *filter* lolos rendah (*lowpass filter*) digunakan untuk menghilangkan *noise*. Beberapa algoritma detektor QRS menggunakan *filter* lolos pita (*bandpass filter*) serta *filter* lolos rendah (*lowpass*) dan lolos tinggi (*highpass*) secara terpisah yang

disusun secara seri untuk mendapatkan komponen-komponen kompleks QRS.

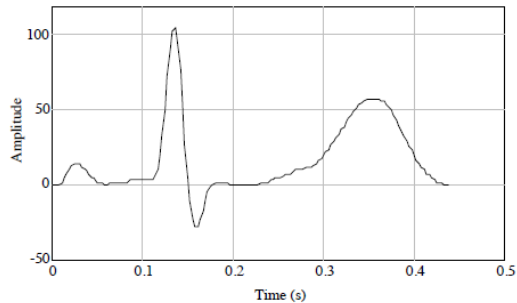


Gambar 2.2 Tahap Penyaringan QRS Detector

Pertama, untuk menipiskan kebisingan, sinyal melewati *filter bandpass* digital yang terdiri dari *filter high pass* dan *low pass*. Kemudian setelah dilakukan penyaringan, dilakukan diferensiasi yaitu pada bentuk sinyal *derivative*. Pada teknik standar ini fungsinya yaitu menemukan lereng tinggi yang biasanya membedakan kompleks QRS dari gelombang EKG lainnya. Selanjutnya diikuti dengan mengkuadratkan hasil dari sinyal *derivative* Informasi tentang kemiringan QRS diperoleh pada tahap turunan (diferensiasi). Transformasi ini berfungsi untuk membuat semua data positif sebelum integrasi berikutnya, dan juga menonjolkan frekuensi yang lebih tinggi pada sinyal yang diperoleh dari proses diferensiasi. Frekuensi yang lebih tinggi ini biasanya karakteristik kompleks QRS. Proses penguadratan mengintensifkan kemiringan kurva respons frekuensi turunan dan membantu membatasi positif palsu yang disebabkan oleh gelombang T dengan energi spektral yang lebih tinggi dari biasanya. Kemudian memindahkan Integrator pergerakan jendela yaitu mendapatkan sinyal *moving window* menghasilkan sinyal yang mencakup informasi tentang kemiringan dan lebar kompleks QRS. Bentuk gelombang kuadrat melewati integrator pergerakan jendela. Integrator ini menghitung area dibawah gelombang kuadrat selama *interval* 150 ms, menghasilkan satu *interval* sampel, dan mengintegrasikan jendela 150 ms baru. Pilih lebar jendela yang cukup panjang untuk memasukkan durasi kompleks QRS abnormal yang panjang, tapi pendek sehingga tidak saling tumpang tindih dengan kompleks QRS dan gelombang T. Ambang batas amplitudo adaptif yang diterapkan pada gelombang yang disaring *bandpass* dan bentuk gelombang integrasi bergerak didasarkan pada perkiraan tingkat sinyal puncak dan kebisingan puncak yang terus diperbarui. Setelah deteksi awal oleh ambang adaptif, proses pengambilan keputusan menentukan akhir apakah kejadian yang

terdeteksi adalah kompleks QRS. Algoritma pengukuran menghitung durasi QRS karena setiap kompleks QRS terdeteksi. Dengan demikian, dua fitur bentuk gelombang tersedia untuk analisis arithmia selanjutnya- *interval R-R* dan durasi QRS.

Filter bandpass untuk algoritma deteksi QRS mengurangi *noise* pada sinyal ECG dengan mencocokkan spektrum kompleks QRS rata-rata. Dengan demikian, hal itu menipiskan kebisingan karena suara otot, gerakan dasar, gangguan 60 Hz dan gangguan gelombang T. Contoh hasil elektrokardiograf seperti Gambar 2.3.



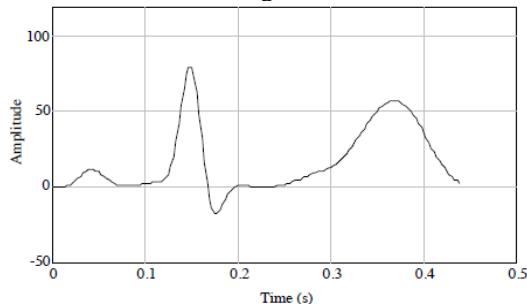
Gambar 2.3 Contoh Elektrokardiograf dengan Sampel 200 Per Detik

2.2.1 Low Pass Filter (LPF)

Persamaan *low pass filter* yaitu seperti Persamaan 2.1 .

$$y(nT) = 2y(nT - T) - y(nT - 2T) + x(nT) - 2x(nT - 6T) + x(nT - 12T) \quad (2.1)$$

Frekuensi *cut-off* pada *low pass filter* ini yaitu 11 Hz. Hasil yang paling nyata adalah redaman kompleks QRS dengan frekuensi lebih tinggi seperti Gambar 2.4. Setiap kebisingan 60 Hz atau suara bising otot juga akan dilemahkan secara signifikan.



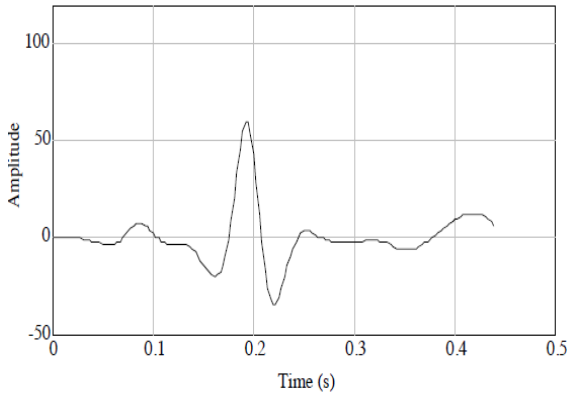
Gambar 2.4 Sinyal Low Pass Filter ECG

2.2.2 High Pass Filter (HPF)

High pass diperoleh dengan membagi *output* dari *filter low-pass* dengan gain dc dan kemudian mengurangkannya dari sinyal aslinya. Fungsi transfer *high-pass filter* seperti Persamaan 2.2.

$$p(nT) = x(nT - 16T) - \frac{1}{32} [y(nT - T) + x(nT) - x(nT - 32T)] \quad (2.2)$$

Frekuensi *cut off* pada *filter* ini yaitu 5 Hz dan gainnya adalah 1. *Filter* ini secara optimal melewati frekuensi karakteristik QRS sambil melemahkan sinyal frekuensi rendah dan lebih tinggi seperti Gambar 2.5.



Gambar 2.5 Sinyal *High Pass Filter* ECG

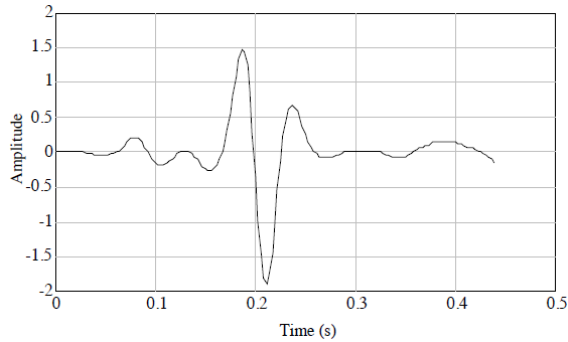
2.2.3 Derivative

Setelah dilakukan penyaringan, dilakukan diferensiasi sehingga didapatkan sinyal *derivative* untuk menemukan lereng tinggi yang biasanya membedakan kompleks QRS dari gelombang EKG lainnya seperti Gambar 2.6.

Persamaan untuk mendapatkan *derivative* yaitu seperti Persamaan 2.3.

$$y(nT) = \frac{2x(nT) + x(nT - T) - x(nT - 3T) - 2x(nT - 4T)}{8} \quad (2.3)$$

Derivative ini mendekati turunan ideal di dc melalui rentang frekuensi 30 Hz.



Gambar 2.6 Sinyal *Derivative*

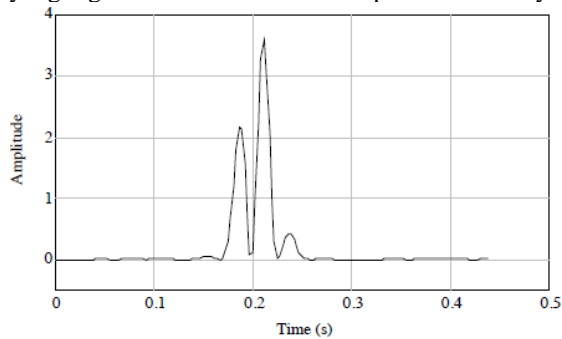
2.2.4 *Squaring*

Proses ini adalah bagian pemrosesan linier detektor QRS. Fungsi kuadrat yang dilalui sinyal sekarang adalah operasi non linier.

Persamaan untuk mendapatkan sinyal *squaring* seperti Persamaan 2.4.

$$y(nT) = [x(nT)]^2 \quad (2.4)$$

Operasi ini membuat semua titik data dalam sinyal yang diproses positif, dan ini memperkuat *output* dari proses derivatif non linear. Ini menekankan frekuensi yang lebih tinggi pada sinyal, yang terutama disebabkan oleh kompleks QRS seperti Gambar 2.7. Fakta yang perlu diperhatikan dalam operasi ini adalah bahwa keluaran dari tahap ini harus dibatasi pada tingkat maksimum tertentu yang sesuai dengan jumlah bit yang digunakan untuk mewakili tipe data dari sinyal.



Gambar 2.7 Sinyal *Squaring*

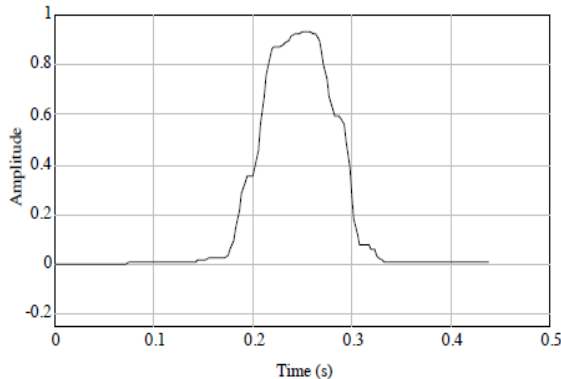
2.2.5 Moving Window

Kemiringan gelombang R saja tidak menjamin cara mendeteksi kejadian QRS. Banyak kompleks QRS yang abnormal yang memiliki amplitudo besar dan durasi yang lama (lereng yang tidak terlalu curam) mungkin tidak terdeteksi dengan menggunakan informasi tentang kemiringan gelombang R saja. Dengan demikian, kita perlu mengekstrak lebih banyak informasi dari sinyal untuk mendeteksi kejadian QRS. Operasi integrasi jendela bergerak selain kemiringan gelombang R. Persamaan untuk mendapatkan sinyal *squaring* seperti Persamaan 2.5.

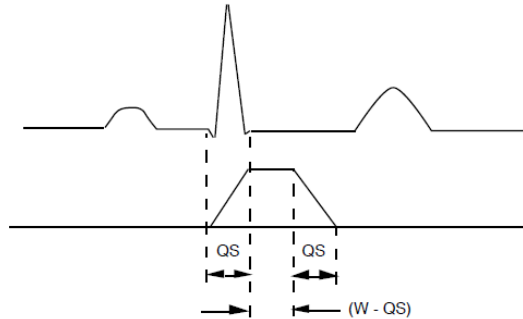
$$y(nT) = \frac{1}{N} [x(nT - (N-1)T) + x(nT - (N-2)T) + \dots + x(nT)] \quad (2.5)$$

Dimana N adalah jumlah sampel di lebar jendela bergerak. Nilai parameter ini harus dipilih dengan hati-hati.

Gambar 2.8 menunjukkan *output* dari integral jendela pemisah untuk sampel ECG. Gambar 2.9 menggambarkan hubungan antara kompleks QRS dan lebar jendela. Lebar jendela kira-kira sama dengan kompleks QRS seluas mungkin. Jika ukuran jendela terlalu besar, bentuk gelombang integrasi akan menggabungkan kompleks QRS dan T secara bersamaan. Di sisi lain, jika ukuran jendela terlalu kecil, kompleks QRS bisa menghasilkan beberapa puncak pada keluaran panggung. Lebar jendela harus dipilih secara eksperimental.



Gambar 2.8 Sinyal *Moving Window*



Gambar 2.9 Hubungan Kompleks QRS dengan Bentuk Gelombang Integrasi Bergerak

2.3 Elektrokardiograf (ECG) [4] [5] [6] [7]

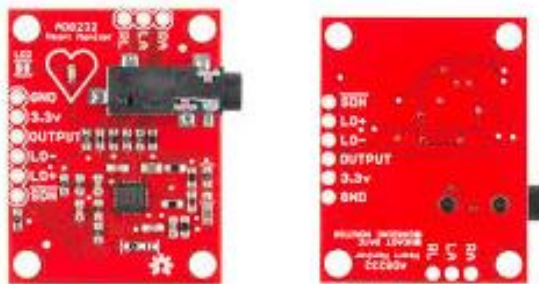
Elektrokardiograf (ECG) adalah hasil diagram yang dibuat oleh sebuah elektrokardiograf, yang merekam aktivitas kelistrikan jantung dalam waktu tertentu. ECG terdiri dari tiga kata yaitu : *elektro*, *kardio*, dan *graf*. Dikatakan elektro karena berkaitan dengan elektronika, *kardio* dari bahasa Yunani yang berarti jantung, dan *graf* dari bahasa Yunani yang berarti menulis. Elektrokardiograf (ECG) merupakan suatu sinyal yang dihasilkan oleh aktivitas listrik otot jantung. Perlu diketahui tubuh manusia menghasilkan listrik walaupun sangat kecil. Sehingga aktivitas otot jantung dapat direkam karena menghasilkan sinyal listrik. ECG ini merupakan rekaman informasi kondisi jantung yang diambil dengan memasang elektroda pada badan. Rekaman ECG ini digunakan oleh dokter atau ahli medis untuk menentukan kondisi jantung dari pasien, yakni untuk mengetahui hal-hal seperti frekuensi (*rate*) jantung, arrhythmia, infark miokard, pembesaran atrium, hipertrofi ventrikular, dll. Sinyal ECG direkam menggunakan perangkat elektrokardiograf.

Elektrokardiograf bekerja dengan prinsip mengukur perbedaan potensial listrik. Seperti yang sudah disebutkan, tubuh manusia menghasilkan listrik walaupun dengan jumlah yang sangat kecil. Apabila ada listrik, maka pasti ada perbedaan potensial atau tegangan listrik. Tegangan listrik ini dapat menggambarkan atau mengilustrasikan keadaan denyut jantung manusia. Pemeriksaan ECG dilakukan dengan menempelkan *lead* (alat penerima impuls listrik jantung) di beberapa lokasi yang telah ditentukan. Setelah itu, informasi mengenai keadaan jantung dapat diketahui melalui pola tampilan sinyal yang dihasilkan.

2.3.1 Modul ECG AD-8232

Modul AD8232 *Heart Rate Monitor* adalah suatu modul yang digunakan untuk mengukur aktivitas listrik jantung. Aktivitas elektrik ini dapat di katakan sebagai ECG yang keluarannya sebagai pembaca sinyal analaog. AD8232 adalah sebuah pengkondisi sinyal terintegrasi untuk EKG dan pengukuran biopotensial lainnya. Modul ini dirancang untuk mengekstrak, memperkuat, dan menyaring sinyal biopotensial kecil ketika dalam keadaan yang memungkinkan terjadi banyak gangguan, seperti yang diciptakan oleh gerak atau penempatan elektroda jarak jauh. Desain ini memungkinkan *ultralow power analog to digital converter* (ADC) atau di hubungkan dengan mikrokontroler untuk memperoleh sinyal *output* dengan mudah. AD8232 dapat menerapkan *filter high pass* dua kutub untuk menghilangkan artifak gerak dan potensial setengah sel elektroda. *Filter* ini digabungkan erat dengan arsitektur instrumentasi penguat untuk mendapatkan hasil kedua gain yang besar dan *high-pass* penyaringan dalam satu tahap, sehingga menghemat ruang dan biaya. Penguat operasional yang tidak terikat dengan AD8232 bisa dibuat sebagai *filter low pass* tiga kutub untuk menghilangkan *noise* tambahan. Pengguna dapat memilih memotong frekuensi dari semua *filter* yang sesuai jenis aplikasi yang diinginkan.

Berdasarkan Gambar 2.10 AD8232 mempunyai sembilan koneksi keluaran dari IC yang dapat dihubungkan dengan SDN, LO+, LO-, *output*, 3,3 V, dan disediakan pin GND untuk mengoperasikan monitor dengan arduino. AD8232 juga dilengkapi dengan pin R (*Right Arm*), L (*Left Arm*) dan F (*Foot*)) untuk menghubungkan dengan sensor elektroda.



Gambar 2.10 AD8232 *Heart Rate Monitor*

Pin AD8232 *Heart Rate Monitor* adalah seperti Tabel 2.2:

Tabel 2.2 Pin AD8232

<i>Board Label</i>	<i>Pin Function</i>
GND	<i>Ground</i>
3,3 V	<i>3,3 V Power supply</i>
<i>Output</i>	<i>Output Signal</i>
LO-	<i>Leads-off Detect -</i>
LO+	<i>Leads-off Detect +</i>
SDN	<i>Shutdown</i>

2.3.2 Sensor Elektroda

Elektroda berdasarkan polaritasnya dibagi menjadi elektroda positif (anode) dan elektroda negatif (katode) dan netral (*ground* elektroda). Fungsi dasar dari elektroda adalah mendeteksi sinyal kelistrikan jantung. Fungsi dari *transducer* adalah untuk mengkonversi informasi biologis menjadi sinyal elektrik yang dapat diukur. *Transducer* ini dipakai dengan menggunakan *interface jelly electrode-electrolyte*. Dengan menggunakan elektroda Ag/AgCl mengurangi *noise* dengan frekuensi rendah pada sinyal ECG yang terjadi karena pergerakan. Agar perolehan gambar ECG jelas, maka dibubuhi krim atau jelly sehingga resistensi antara elektroda dengan kulit menjadi seminimal mungkin seperti Gambar 2.11.



Gambar 2.11 Sensor Elektroda

2.3.3 Monitoring Sinyal ECG

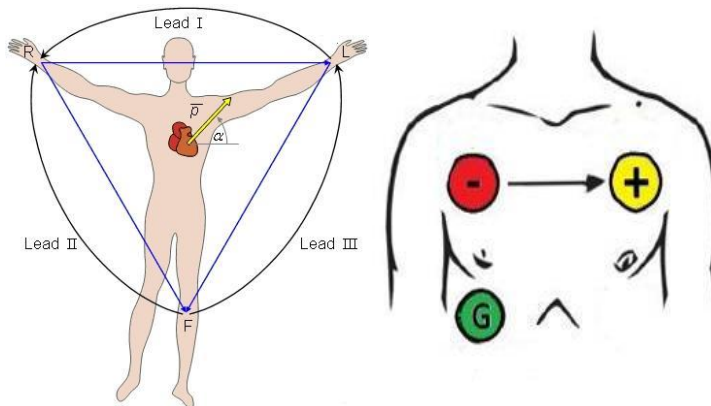
Dalam memonitoring sinyal ECG biasanya digunakan metode yaitu teknik monitoring standar ekstremitas (metoda *Einthoven*) atau *standard limb leads* dan teknik monitoring tambahan atau *augmented limb leads* seperti Gambar 2.12.

Teknik monitoring standar ekstremitas (metoda *Einthoven*) dilakukan di 3 tempat monitoring yaitu:

1. *Lead I* dibentuk dengan membuat lengan kiri (LA – *left arm*) elektroda positif dan lengan kanan (RA – *right arm*) elektroda negatif. Sudut orientasi 0° .
2. *Lead II* dibentuk dengan membuat kaki kiri (LL – *left leg*) elektroda positif dan lengan kanan (RA – *right arm*) elektroda negatif. Sudut orientasi 60° .
3. *Lead III* dibentuk dengan membuat kaki kiri (LL – *left leg*) elektroda positif dan lengan kiri (LA – *left arm*) elektroda negatif. Sudut orientasi 120° .

Teknik monitoring tambahan atau *augmented limb leads* dilakukan di 3 tempat yaitu:

1. aVL dibentuk dengan membuat lengan kiri (LA – *left arm*) elektroda positif dan anggota tubuh lainnya (ekstremitas) elektroda negatif. Sudut orientasi -30° .
2. aVR dibentuk dengan membuat lengan kanan (RA – *right arm*) elektroda positif dan anggota tubuh lainnya (ekstremitas) elektroda negatif. Sudut orientasi -150° .
3. aVF dibentuk dengan membuat kaki kiri (LL – *left leg*) elektroda positif dan anggota tubuh lainnya (ekstremitas) elektroda negatif. Sudut orientasi 90° monitoring EKG *prekordia*/dada atau *standard chest leads* monitoring.



Gambar 2.12 Konfigurasi Pemasangan Elektroda

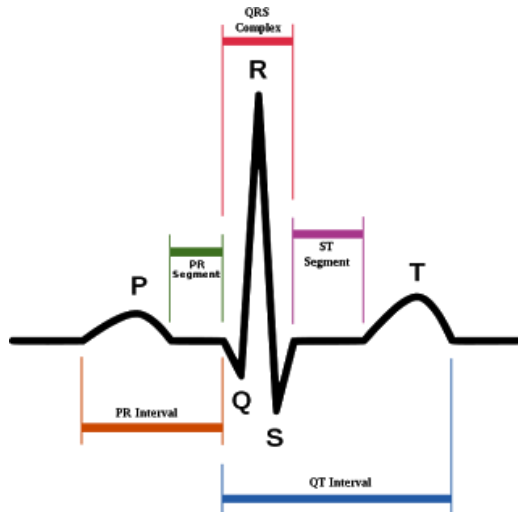
2.3.4 Gelombang Elektrokardiogram (ECG)

Sinyal ECG terdiri dari gelombang P, kompleks QRS, dan gelombang T seperti Gambar 2.14 yang digunakan untuk mendeteksi kelainan jantung atau aritmia (*arrythmia*). Urutan terjadinya sinyal EKG yang dapat menimbulkan gelombang P, kompleks QRS, dan gelombang T seperti Gambar 2.13:

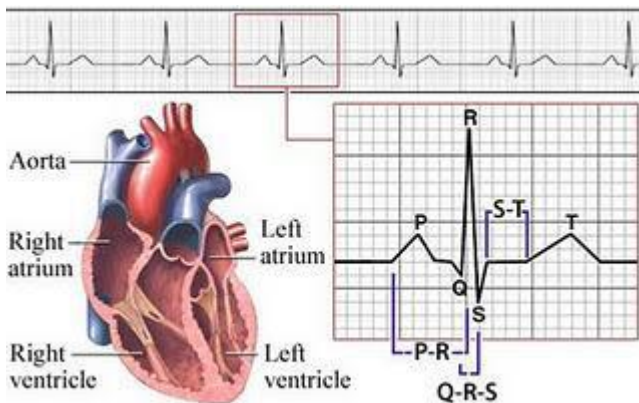
1. Setiap siklus kontraksi dan relaksasi jantung dimulai dengan depolarisasi spontan pada nodus. Peristiwa ini tidak tampak pada rekaman EKG
2. Gelombang P merekam peristiwa depolarisasi dan kontraksi atrium (*atria contract*). Bagian pertama gelombang P menggambarkan aktivitas atrium kanan; bagian kedua mencerminkan aktivitas atrium kiri.
3. Komplek QRS: mewakili depolarisasi otot miokardium (kontraksi ventrikel). Amplitudo pada QRS kompleks memiliki rentang nilai normal yg beragam, tetapi dianggap tinggi abnormal (*abnormally high*) jika amplitudo lebih besar daripada 1,1 mV (11 mm) pada *lead aVL*, lebih besar daripada 2 mV (20 mm) pada *lead aVF* di *Frontal Plane Lead* atau lebih besar dari pada 3 mV (30 mm) pada *lead precordial (Horizontal Plane)*. Untuk durasi (panjang waktu) Komplek QRS memiliki rentang normal dari 50-100 ms (0,05-0,10 sec) atau 1 - 2 setengah kotak kecil .

Kesimpulan untuk tatanama Gelombang di kompleks QRS.

- a. Gelombang Q: Defleksi negatif pertama setelah gelombang P
 - b. Gelombang R: Defleksi positif pertama setelah gelombang P
 - c. Gelombang S: Defleksi negatif kedua setelah gelombang P (bila ada gelombang Q) atau defleksi negatif pertama setelah gelombang R.
4. Gelombang T: menggambarkan repolarisasi otot ventrikel (relaksasi ventrikel). Amplitudo gelombang T diukur dari garis dasar (isoelektrik) hingga ke titik tertinggi gelombang, dengan nilai tidak melebihi 0,5 mV (5 mm = 5 kotak kecil) di seluruh *frontal Plane Lead* atau 1 mV (10 mm) di seluruh *precordial lead (horizontal plane lead)*. Polaritas gelombang T bervariasi tergantung *lead*, normalnya positif (melengkung ke atas) pada *lead I, II, dan V3-V6* pada dewasa dan negatif (*inverted*) pada *lead aVR*, dan beragam pada *lead III, aVL, aVF, dan V1-V2*.



Gambar 2.13 Gelombang *Output* ECG



Gambar 2.14 Gelombang Sinyal ECG

2.4 USB UART FT232RL [8]

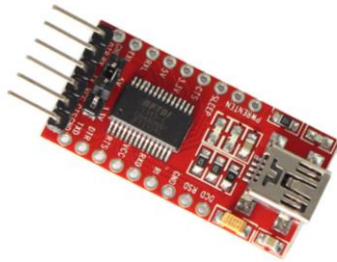
Modul FT232RL adalah modul konversi sinyal USB ke sinyal TTL/UART (*USB-to-TTL Converter*) yang andal dan praktis untuk digunakan pada rangkaian elektronika berbasis mikrokontroler, dengan

demikian perangkat elektronika buatan. Berdasarkan Gambar 2.15 Modul ini berkomunikasi dengan perangkat lain lewat komunikasi standar USB. Konektor USB yang terpasang adalah konektor tipe mini-B *female*. Modul FT232 berfungsi untuk merubah data USB yang berasal dari port USB menjadi data serial dengan level tegangan TTL. Sehingga dengan menggunakan modul FT232 ini komunikasi data serial (UART) melalui port USB dapat dilakukan. Semua proses handshaking, enumerasi dan lain-lain yang diperlukan agar dapat menggunakan port USB telah ditangani oleh modul FT232 tersebut, sehingga memudahkan pengguna untuk menggunakannya. Pin pada *board* ini cocok dengan kabel FTDI yang umum digunakan untuk memprogram *board* Arduino yang asli maupun yang kompatibel.

USB ini juga dapat digunakan pada aplikasi umum yang menggunakan koneksi serial. Perbedaan utama dari *board* ini dengan yang sejenis adalah digunakannya pin DTR dan bukannya pin RTS seperti pada kabel FTDI. Pin DTR ini memungkinkan *board* Arduino yang diprogram untuk melakukan *auto reset* setiap kali dimasukkannya *sketch* (program) kedalam Arduino. *Board* ini mempunyai LED TX dan RX yang memungkinkan kita untuk melihat lalu lintas serial data melalui LED tersebut, sehingga dapat diketahui apakah *board* ini dapat bekerja dengan baik, tapi *board* ini memerlukan kabel mini USB untuk menghubungkan dengan komputer. Kabel FTDI sangat terlindungi dari gangguan, tetapi ukurannya terlalu besar sehingga tidak dapat ditanam ke dalam proyek Arduino anda dengan mudah.



Gambar 2.15 Konfigurasi Pin FT232RL



Gambar 2.16 *Board FT323RL*

Spesifikasi dari Gambar 2.16 adalah sebagai berikut :

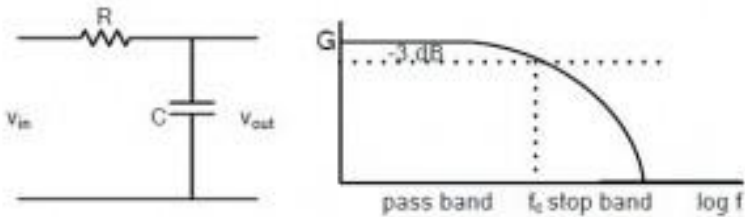
1. Pin *output*:
 - a. DTR
 - b. RX
 - c. TX
 - d. VCC
 - e. CTS
 - f. GND
2. Pilihan tegangan: 3,3 V atau 5 V
3. LED indikator RX dan TX
4. Bisa dikoneksikan langsung ke UART Arduino

2.5 Rangkaian Low Pass Filter [9]

Filter adalah sebuah rangkaian yang dirancang agar mengalirkan suatu pita frekuensi tertentu dan menghilangkan frekuensi yang berbeda dengan pita ini. Istilah lain dari *filter* adalah rangkaian yang dapat memilih frekuensi agar dapat mengalirkan frekuensi yang diinginkan dan menahan, atau membuang frekuensi yang lain. Jaringan *filter* bisa bersifat aktif maupun pasif. Perbedaan dari komponen aktif dan pasif adalah pada komponen aktif dibutuhkan sumber agar dapat bekerja (op-amp dan transistor membutuhkan sumber lagi agar dapat bekerja/digunakan), sedangkan komponen pasif tidak membutuhkan sumber lagi untuk digunakan atau bekerja. (Chattopadhyay.1989). Dibidang elektronika khususnya untuk elektronika analog, *Filter* sering digunakan untuk meloloskan frekuensi yang dikehendaki atau menahan frekuensi yang tidak dikehendaki. *Filter* yang digunakan biasanya terdiri dari tiga macam konfigurasi yang dapat dibagi sebagai berikut :

1. *Low Pass Filter* (LPF)
2. *High Pass Filter* (HPF)
3. *Band Pass Filter* (BPF)
4. *Band Stop Filter* (BSF)

Low Pass Filter adalah *filter* yang berfungsi meneruskan sinyal *input* yang frekuensinya berada dibawah frekuensi tertentu, diatas frekuensi tersebut (frekuensi *cut-off*) sinyal diredam (FcoL). Rangkaian *low pass filter* dapat dibangun menggunakan dua jenis rangkaian dasar, yakni rangkaian *low pass filter* induktif rangkaian *low pass filter* kapasitif. Untuk rangkaian *low pass filter* induktif, rangkaian terdiri dari induktor (L) dan beban (R) sedangkan rangkaian *low pass filter* kapasitif dibangun menggunakan dua komponen utama yaitu resistor (R) dan kapasitor (C). Rangkaian *low pass filter* RC merupakan jenis *filter* pasif, dengan respon frekuensi yang ditentukan oleh konfigurasi R dan C yang digunakan. Rangkaian dasar LPF dan respon frekuensi LPF seperti Gambar 2.17.



Gambar 2.17 Rangkaian Dasar dan Respon Frekuensi *Low Pass Filter* RC

Frekuensi *cut-off* (f_c) dari *filter* pasif lolos bawah (*Low Pass Filter*, LPF) dengan RC dapat dituliskan dalam persamaan matematik seperti Persamaan 2.6.

$$f_c = \frac{1}{2\pi RC} \quad (2.6)$$

Rangkaian *filter* pasif LPF RC terlihat seperti pembagi tegangan menggunakan R. Dimana pada *filter* LPF RC ini tegangan *output* diambil pada titik pertemuan RC. Tegangan *output* (V_{out}) *filter* pasif LPF seperti terlihat pada rangkaian LPF dapat diekspresikan dalam persamaan matematis seperti Persamaan 2.7.

$$V_{out} = \frac{1/j\omega C}{1/j\omega C + R} \cdot V_{in} \quad (2.7)$$

Besarnya penguatan tegangan (G) pada *filter* pasif yang ideal maksimum adalah 1 = 0dB yang hanya terjadi pada frekuensi sinyal *input* dibawah frekuensi *cut-off* (f_c). Penguatan tegangan (G) *filter* LPF RC pasif dapat dituliskan dalam persamaan matematis seperti Persamaan 2.8.

$$G = \left| \frac{V_{out}}{V_{in}} \right| \quad (2.8)$$

Penguatan tegangan (G) LPF RC dapat dituliskan dalam satuan dB seperti Persamaan 2.9.

$$G = 20 \log \frac{V_{out}}{V_{in}} = 20 \log \frac{1}{1 + \omega^2 C^2 R^2} \quad (2.9)$$

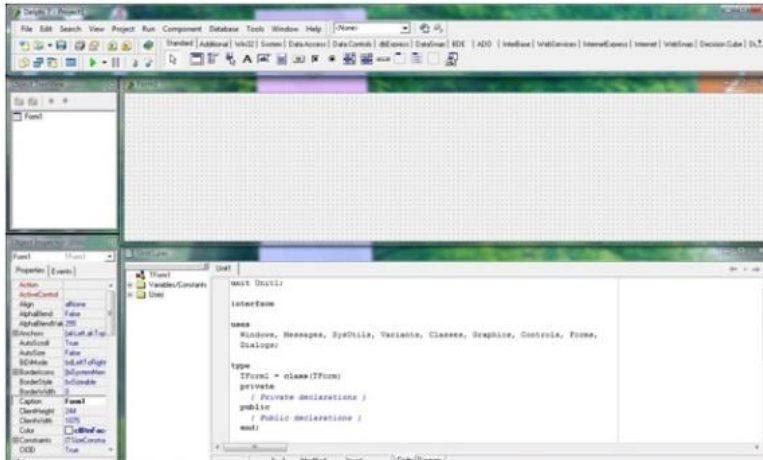
Pada *filter* lolos rendah (*low pass filter*, LPF) terdapat beberapa karakteristik mendasar sebagai berikut.

- Pada saat frekuensi sinyal *input* lebih rendah dari frekuensi *cut-off* (f_c) ($f_{in} \ll f_c$) maka penguatan tegangan / Gain (G) = 1 atau G=0dB.
- Pada saat frekuensi sinyal *input* sama dengan frekuensi *cut-off* (f_c) ($f_{in} = f_c$) maka $\omega = 1/RC$ sehingga penguatan tegangan / Gain (G) menjadi -3 dB atau terjadi pelemahan tegangan sebesar 3 dB.
- Pada saat frekuensi sinyal *input* lebih tinggi dari frekuensi *cut-off* (f_c) ($f_{in} \gg f_c$) maka besarnya penguatan tegangan (G) = $1/\omega RC$ atau G = -20 log ωRC
- Sehingga dapat ditarik kesimpulan bahwa *filter* lolos rendah (*Low pass Filter*, LPF) hanya meloloskan sinyal dengan frekuensi yang lebih rendah dari frekuensi *cut-off* (f_c) saja.

2.6 Software Delphi [10]

Delphi adalah sebuah IDE Compiler untuk bahasa pemrograman Pascal dan lingkungan pengembangan perangkat lunak yang digunakan untuk merancang suatu aplikasi program. Delphi merupakan salah satu bahasa pemrograman berbasis visual yang digunakan untuk membuat program aplikasi pada komputer seperti *visual basic*. Bahasa pemrograman yang digunakan merupakan turunan dari bahasa pemrograman pascal jadi untuk mereka yang sudah mengenal pascal sangat membantu sekali dalam penggunaan Delphi. IDE (*Integrated Development Environment*) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak.

Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. IDE atau *Integrated Development Environment* merupakan bahasa pemrograman dimana proses program berjalan atau dilakukan dalam satu jendela seperti Gambar 2.18



Gambar 2.18 Jendela *Software* Delphi

IDE Delphi tersebut dijelaskan secara garis besar dari yang tampak pada Gambar 2.18 yaitu sebagai berikut :

1. *Menubar*



Gambar 2.19 *Menubar* Delphi

2. *Toolbar*



Gambar 2.20 *Toolbar* Delphi

Berdasarkan Gambar 2.19 dan Gambar 2.20, *Menubar* dan *Toolbar* tersebut hampir sama dengan aplikasi yang biasa kita temui pada aplikasi-aplikasi *windows* lain. Dari *File*, *Edit*, *Search*, *View*,

Project, Component, Database, Tools, Window, Help, dimana *Run* berfungsi untuk membuka hasil dari aplikasi yang sudah kamu buat.

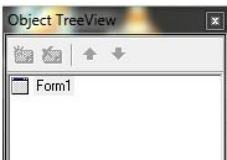
3. *Component Palette*



Gambar 2.21 *Component Delphi*

Component Palette ini berisi *visual component library* yang merupakan pustaka yang kesemuanya dilambangkan dengan *icon-icon*. Seperti Gambar 2.21. *Icon-icon* tersebut dikelompokkan ke dalam tab-tab yang bisa memudahkan dalam memilih.

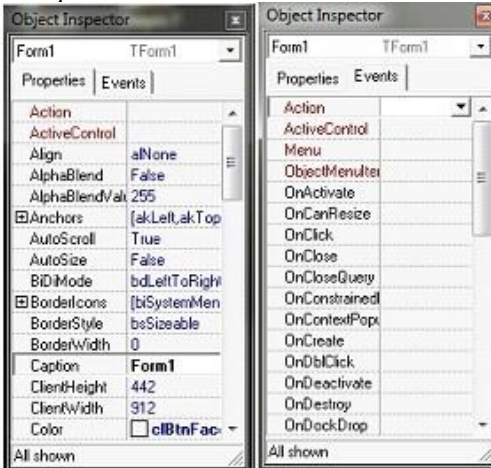
4. *Object Tree View*



Gambar 2.22 *Object Tree View Delphi*

Berdasarkan Gambar 2.22, *Object Tree View* merupakan daftar komponen yang digunakan *programmer* dimana akan memudahkan *programmer* dalam menunjuk komponen tertentu terutama pada banyak cabang yang dibuat.

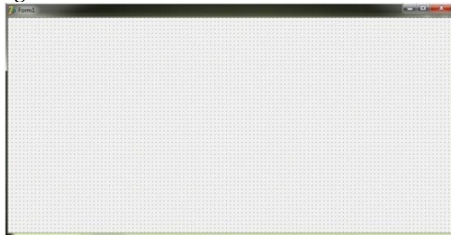
5. *Object Inspector*



Gambar 2.23 *Object Inspector Delphi*

Berdasarkan Gambar 2.23, *Object Inspector* dibagi menjadi dua tab yaitu tab *Properties* dan tab *Event*. Tab *Properties* digunakan untuk memanipulasi properti yang dimiliki oleh komponen tertentu, misalkan ukuran, warna dan *caption* dari komponen. Sedangkan Tab *Event* digunakan untuk menangani pemasukan kode pada kejadian tertentu dari suatu komponen, misalnya : kejadian ketika komponen *button* (tombol) diklik atau *on click*.

6. *Form Designer*



Gambar 2.24 *Form Designer* Delphi

Form Designer berfungsi sebagai tempat untuk mendesain *form* untuk aplikasi yang akan dibuat, dan juga sebagai tempat untuk meletakkan komponen-komponen yang kita ambil dari *component palette*. Tampilan *form designer* seperti Gambar 2.24.

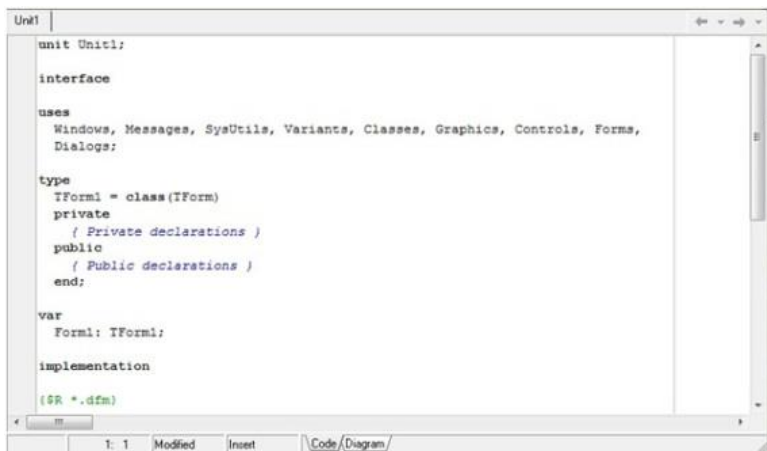
7. *Code Explorer*



Gambar 2.25 *Code Explorer* Delphi

Code Explorer ini digunakan untuk menampilkan seluruh variabel, tipe, dan *routine* yang didefinisikan pada sebuah unit. Tampilan *Code Explorer* Delphi seperti Gambar 2.25.

8. Code Editor



Gambar 2.26 Gambar *Code Editor* Delphi

Code Editor berfungsi menuliskan kode program. Pada bagian *code editor* terdapat 3 bagian utama yaitu, bagian paling kiri yang berisi berupa angka menunjukkan baris dan kolom. keterangan *modified* menunjukkan bahwa telah terjadi modifikasi terhadap baris program dan paling kanan menunjukkan status *keyboard* tentang tombol *insert* atau *over write*. Tampilan *Code Explorer* Delphi seperti Gambar 2.26.

2.7 Jantung [11] [12]

Organ jantung adalah organ berdenyut yang memompa darah ke seluruh tubuh. Rata-rata, jantung beratnya sekitar 10,5 ons (300 gram). suatu organ otot berongga yang terletak di pusat dada. Bagian kanan dan kiri jantung masing memiliki ruang sebelah atas (atrium) yang mengumpulkan darah dan ruang sebelah bawah (ventrikel) yang mengeluarkan darah. Agar darah hanya mengalir dalam satu arah, maka ventrikel memiliki satu katup pada jalan masuk dan satu katup pada jalan keluar. Kecepatan normal jantung biasanya 60 sampai 100 detak per menit. Namun demikian, detak jantung normal lebih rendah untuk usia muda, khususnya yang mempunyai fisik fit. Jantung hampir sepenuhnya diselubungi oleh paru-paru, tertutup oleh selaput ganda

yang bernama perikardium, yang tertempel pada diafragma. Lapisan pertama menempel sangat erat pada jantung, sedangkan lapisan luarnya lebih longgar dan berair, untuk menghindari gesekan antar organ dalam tubuh yang terjadi karena gerakan memompa konstan jantung. Aliran darah di dalam jantung diuraikan sebagai berikut:

1. Darah dari pembuluh balik memasuki serambi jantung kanan dan kiri.
2. Serambi mulai memompa atau menekan darah keluar menuju bilik. Saat itu serambi berkontraksi.
3. Ketika serambi berkontraksi, bilik kanan dan kiri relaksasi (tidak memompa). Saat itu bilik menerima darah dari serambi.
4. Bilik kanan dan kiri kemudian berkontraksi menekan darah ke dalam dua arteri besar menuju tubuh dan paru-paru.
5. Saat bilik memompa darah ke pembuluh nadi, serambi Saat itu darah dari pembuluh balik (dari tubuh) kembali memasuki serambi, seperti proses nomor 1 dan siklus jantung terulang lagi.

Setiap hari atau setiap saat jantung tidak pernah berhenti bekerja, walaupun saat tidur. Setiap kontraksi jantung memaksa sekitar 2,5 ons (74 mililiter) dari darah ke dalam aliran darah. Ini menambahkan sampai sekitar 10 liter (4,7 liter) darah setiap menit. Sebuah jantung rata-rata akan memompa sekitar 1.800 galon (6.800 liter) darah setiap hari. Dengan berolahraga, jumlah yang dapat meningkatkan sebanyak enam kali. Dalam seumur hidup rata-rata, jantung akan memompa sekitar 100 juta galon (380 juta liter) darah. Adapun fungsi organ jantung pada tubuh manusia adalah memompa darah ke seluruh tubuh, kembali ke jantung dan menampungnya kembali setelah darah dibersihkan kembali oleh organ paru-paru. Fungsi utama jantung adalah menyediakan oksigen (O_2) ke seluruh tubuh dan membersihkan tubuh dari hasil metabolisme berupa karbondioksida (CO_2). Jantung melaksanakan fungsi tersebut dengan mengumpulkan darah yang kekurangan oksigen dari seluruh tubuh dan memompanya ke dalam paru-paru, dimana darah akan mengambil oksigen dan membuang karbondioksida (disebut sirkulasi pulmonal). Kemudian jantung mengumpulkan darah yang kaya oksigen dari paru-paru dan memompanya ke jaringan di seluruh tubuh (disebut sirkulasi sistemik).

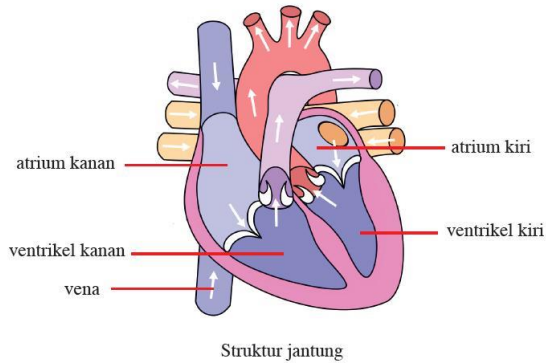
Pada saat berdenyut, setiap ruang jantung melakukan relaksasi dan pengisian darah pada jantung (disebut periode diastol). Sebaliknya jantung berkontraksi untuk mengosongkan isinya (disebut periode sistol). Kedua serambi (atrium) mengendur dan berkontraksi secara

bersamaan, dan kedua bilik (ventrikel) juga mengendur dan berkontraksi secara bersamaan pula untuk melakukan mekanisme tersebut. Ketika melakukan kontraksi, jantung menjadi berdenyut secara berirama, hal ini akibat dari adanya potensial aksi yang ditimbulkan oleh kegiatan diri jantung itu sendiri. Kejadian tersebut diakibatkan karena jantung memiliki sebuah mekanisme untuk mengalirkan listrik yang ditimbulkannya sendiri untuk melakukan kontraksi atau memompa dan melakukan relaksasi atau dikenal dengan istilah sistem listrik jantung. Mekanisme aliran listrik yang menimbulkan aksi tersebut dipengaruhi oleh beberapa jenis elektrolit seperti K^+ , Na^+ , dan Ca^{2+} . Sehingga apabila didalam tubuh terjadi gangguan pada kadar elektrolit tersebut maka akan menimbulkan gangguan pula pada mekanisme aliran listrik pada jantung manusia.

Berdasarkan Gambar 2.27, Jantung terdiri atas empat ruang yaitu serambi kanan, serambi kiri, bilik kanan, dan bilik kanan. Atrium adalah ruangan sebelah atas jantung dan berdinding tipis, sedangkan ventrikel adalah ruangan sebelah bawah jantung dan mempunyai dinding lebih tebal karena harus memompa darah ke seluruh tubuh. Berikut ini fungsi masing-masing ruang yang ada pada jantung:

1. Serambi (atrium) kanan berfungsi sebagai penampung darah rendah oksigen (O_2) dari seluruh tubuh.
2. Serambi (atrium) kiri berfungsi menerima darah yang kaya oksigen dari paru-paru dan mengalirkan darah tersebut ke paru-paru.
3. Bilik (ventrikel) kanan berfungsi menerima darah dari atrium kanan dan memompakannya ke paru-paru.
4. Bilik (ventrikel) kiri berfungsi untuk memompakan darah yang kaya oksigen (O_2) keseluruh tubuh.

Pada masing-masing bagian jantung, atrium dan ventrikel dipisahkan oleh suatu sekat/septum. Kedua atrium dipisahkan oleh suatu sekat antar atrium (septum interatriorum), sementara kedua ventrikel dipisahkan oleh sekat antar ventrikel.



Gambar 2.27 Struktur Jantung

2.8 Arithmia [13] [14]

Aritmia adalah masalah pada irama jantung ketika organ tersebut berdetak terlalu cepat, terlalu lambat, atau tidak teratur. Aritmia terjadi karena impuls elektrik yang berfungsi mengatur detak jantung tidak bekerja dengan baik. Gerakan jantung tersebut memiliki irama. Irama yang normal untuk jantung adalah dalam satu kali berkontraksi terdiri dari satu kali kontraksi atrium dan satu kali kontraksi bilik. Bila terjadi gangguan irama, misal dua kali kontraksi atrium dan satu kali kontraksi bilik atau satu kali kontraksi atrium kemudian ada jeda dahulu baru terjadi kontraksi bilik, akan menyebabkan darah yang dipompakan jantung tidak optimal. Kecepatan jantung yang bervariasi adalah normal. Kecepatan jantung tidak hanya merespon kegiatan olah raga dan tidak beraktivitas tetapi juga dipengaruhi oleh nyeri dan emosi. Hanya ketika kecepatan jantung tidak wajar cepatnya (takikardi) atau lambatnya (bradikardi) atau ketika impuls elektrik bergerak dalam aliran yang abnormal sehingga detak jantung diperkirakan mempunyai irama yang abnormal (aritmia). Irama abnormal mungkin teratur atau tidak teratur. Jantung yang berdenyut melambat tentu akan mengganggu aliran darah sampai ke otak sehingga penderitanya sewaktu-waktu dapat pingsan. Sebaliknya, jika jantung berdenyut terlalu cepat dalam jangka yang lama maka dapat mengarah pada gagal jantung kongestif yang tentunya sangat berbahaya. Aritmia cepat (takikardi) bisa disebabkan oleh olah raga, stress emosional, konsumsi alkohol berlebih, merokok, atau menggunakan obat yang mengandung *stimulant*, seperti obat flu dan demam. Aritmia lambat (bradikardi) dapat disebabkan oleh nyeri, lapar,

pegal, gangguan pencernaan (seperti diare dan muntah), atau kembung, yang dapat merangsang saraf vagus berlebihan (dengan stimulasi yang cukup, walau jarang terjadi, saraf vagus dapat menyebabkan jantung berhenti) dalam sebagian besar keadaan, aritmia dapat selesai dengan sendirinya

Pada kelainan jantung ada yang namanya arithmia dan kelainan bentuk. Macam-macam arithmia adalah sebagai berikut:

1. Sinus takikardi yaitu meningkatnya aktifitas nodus sinus, gambaran yang penting pada ECG adalah laju gelombang lebih dari 100 kali per menit.
2. Sinus bradikardi yaitu penurunan laju depolarisasi atrium. Gambaran yang terpenting pada ECG adalah laju kurang dari 60 per menit. Irama teratur RR *interval* jaraknya sama. PP *interval* jaraknya sama. Komplek QRS harus sama. Impuls dari SA node yang ditandai dengan adanya gel P yang mempunyai bentuk sama.
3. Komplek atrium prematur yaitu impul listrik yang berasal di atrium tetapi di luar nodus sinus menyebabkan kompleks atrium prematur, timbulnya sebelum denyut sinus berikutnya. Gambaran ECG menunjukkan irama tidak teratur, terlihat gelombang P yang berbeda bentuknya dengan gelombang P berikutnya.
4. Takikardi atrium yaitu biasanya diawali oleh suatu kompleks atrium prematur sehingga terjadi resenstri pada tingkat nodus AV.
5. Flutter atrium yaitu resenstri pada tingkat atrium. Depolarisasi atrium cepat dan teratur, dan gambarannya terlihat terbalik.
6. Fibrilasi atrium bisa timbul dari fokus ektopik ganda dan atau daerah resenstrimultipel. Aktifitas atrium sangat cepat.
7. Fibrilasi ventrikular terjadi apabila impuls elektrik yang kacau dan cepat menyebabkan ventrikel hanya bergetar saja dan tidak benar-benar memompa darah ke seluruh tubuh.
8. Takikardi ventrikuler yaitu detak jantung diatas normal yang disebabkan sinyal elektrik yang terbagi dua antara menuju atrium dan menuju ventrikel jantung. Detak jantung yang terlalu cepat ini membuat ventrikel tidak bisa terisi dan berkontraksi secara efisien dalam memompa darah.

Berikut beberapa kelainan pada anatomi (bentuk) dari jantung:

1. Demam Rematik Akut (DRA)

Demam reumatik adalah suatu penyakit peradangan pada jantung yang mana dapat mengenai pada semua lapisan jantung mulai dari endokardium, miokardium dan perikardium. Dimana penyebab

demam reumatik ini adalah virus streptococcus beta hemoliticus grup A disalurkan nafas bagian atas (ISPA). Dimana gejala dari demam reumatik ini adalah penderita merasakan panas tinggi, adanya pembengkakan pada persendian besar, misalnya: lutut, bahu, siku, pergelangan tangan atau kaki disertai warna kemerahan dan terasa panas. Biasanya penyakit ini mulai menyerang pada anak-anak yang berumur 5-12 tahun, yang kira-kira 7-15 hari atau lebih lama sebelum terserang oleh demam reumatik akut telah menderita infeksi oleh jenis bakteri Steptokokus hemolitikus yang menyerang bagian tekak, hidung, atau telinga atau ditempat lain.

2. Penyakit Jantung Akibat Gangguan Peredaran Darah (GPD)
Penyakit jantung ini disebabkan oleh hal dibawah ini:
 - a. Penyakit jantung hipertensif: pembesaran jantung kiri akibat tekanan darah yang meningkat secara terus menerus.
 - b. Penyakit jantung akibat diperlukannya karbon dikosida yang besar dari jantung, misalnya pada penyakit anemia yang parah, aktivitas kelenjar gondok yang meninggi, penyakit paru-paru menahun, dan lain-lain.
3. Penyakit karena infeksi oleh kuman-kuman yang ganas
Pada penyakit infeksi selalu ada kuman-kuman penyebab penyakit yang akan terbawa ke dalam peredaran darah. Dimana kuman-kuman tersebut dapat menyebabkan radang pada organ maupun jaringan yang dilalui darah, seperti pada organ jantung.
4. *Aterosklerose*
Aterosklerose adalah suatu penyakit degeneratif pada arteri yang menyebabkan dinding arteri menjadi keras, antara lain akibat terjadinya pengapuran. Dimana sebagian besar *aterosklerose* ini disebabkan oleh makanan yang banyak mengandung banyak kolesterol dan lemak hewani.
5. Kelainan Jantung Fungsional
Penyakit ini muncul oleh karena sistem saraf otonom yang sedang tegang. Dimana ketegangan ini dapat menyebabkan nyeri area dada, napas sesak, jantung berdebar-debar, bahkan ada yang mengatakan jantungnya seperti mau berhenti. Adapun sebab kelainan jantung fungsional ini adalah: kurang tidur, kebanyakan minum kopi, mengisap rokok, stres maupun alergi terhadap makanan.
6. Penyakit Jantung Bawaan (PJB) atau Kelainan Jantung Kongenital (KJK)

Penyakit ini biasanya sudah ada sejak bayi dilahirkan, sebagai akibat dari terganggunya perkembangan jantung janin pada trimester pertama saat masa kehamilan.

BAB III

PERANCANGAN SISTEM

Pada bab ini dibahas mengenai pembuatan instrument dan *software* untuk tugas akhir ini. Pada pembuatan instrument yang dibahas yaitu mengenai perancangan instrument dengan menghubungkan Sensor ECG (Elektroda) ke Modul ECG (AD8232) kemudian dari keluaran modul di *filter* dengan menggunakan *Low Pass Filter* yang nantinya dihubungkan ke Mikrokontroler Arduino UNO yang menggunakan USB FT232RL, kemudian di terjemahkan ke PC. Sedangkan dalam perancangan *software*, yaitu membuat program pembacaan sensor ECG modul AD8232 pada Mikrokontroler, membuat program untuk menghasilkan nilai data *integer* dari Mikrokontroler. Kemudian membuat program pembacaan sensor ECG yang ditampilkan pada Delphi berupa tampilan sinyal dan nilai *setting point* dengan menerjemahkan nilai *integer* dari Mikrokontroler. Setelah itu membuat program untuk menampilkan sinyal dari pengambilan data hasil sensor ECG pada Delphi. Hasil yang ditampilkan nanti berupa sinyal ECG, sinyal LPF, sinyal *squaring*, sinyal *derivative*, sinyal *moving window*. Sehingga dalam pengukuran ECG lebih akurat dan mempermudah dalam mendeteksi Arithmia.

3.1 Perancangan Elektronik

Pada sub bab ini dibahas mengenai *setting point* mikrokontroler, rangkaian *low pass filter*, perancangan modul AD8232 dihubungkan Arduino UNO. perancangan USB FT232RL dihubungkan Arduino UNO.

Instrument detektor Arithmia ini menggunakan sensor elektroda sebagai sensor elektrodioGRAF yang mana *lead I* berwarna merah adalah *lead R (Right Arm)* yang ditempelkan pada pergelangan tangan sebelah kanan, *lead II* berwarna kuning adalah *lead L (Left Arm)* yang ditempelkan pada pergelangan tangan sebelah kiri, *lead III* berwarna hijau adalah *F (Feet)* yang ditempelkan pada kaki sebelah kanan. Kemudian probe dari sensor ECG dihubungkan ke *board* AD8232 (modul ECG).

Setelah itu dari *Board* AD8232 hasil *output* analog masuk ke *Low pass Filter* terlebih kemudian pin LO+ dan LO- serta VCC 3,3 V masuk ke pin arduino. Terjadilah frekuensi sampling 200 Hz. Arduino langsung dihubungkan dengan serial COM untuk memasukkan program arduino.

Kemudian serial COM diganti dengan USB TTL FT232RL sebagai serialnya. Sumber dari instrument ini yaitu langsung dari Laptop PC. Laptop tidak boleh dicass saat menggunakan instrument ini karena bisa mengganggu stabilitas tegangan yang menjadi sumber alat sehingga *output* yang keluar dari alat tidak akan terdeteksi hasil yang bagus

Cara menampilkan ke PC menggunakan *software* borland delphi. Hasil yang diampilkkan nantinya menjadi suatu refrensi pendeteksian arithmia melalui pendeteksian sinyal QRS.

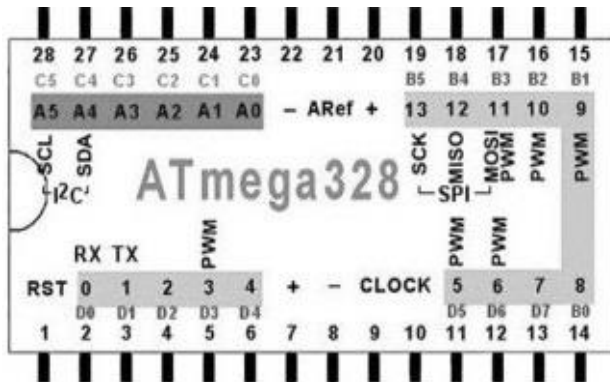
3.1.1 Setting Point Mikrokontroler

Mikrokontroler digunakan sebagai pusat kendali pada instrumentasi *detector* arithmia ini. Mikrokontroler yang digunakan adalah mikrokontroler Arduino UNO. Arduino UNO adalah sebuah *board* mikrokontroler didasarkan pada ATmega 328. Pada instrument ini digunakan beberapa pin mikrokontroler arduino UNO seperti Gambar 3.1.

Berdasarkan Tabel 3.1, Pin A0 pada arduino disetting sebagai *output* modul ECG. Kemudian pin 3 diatur sebagai Rx pada arduino yang nantinya dihubungkan USB FT232RL. Pin 4 diatur sebagai Tx pada arduino yang dihubungkan dengan USB FT232RL. Pin 10 sebagai *input* dari LO+ modul AD8232 yang dibaca digital oleh mikrokontroler. Pin 11 sebagai *input* LO- modul AD8232 yang dibaca digital oleh mikrokontroler. Pin Areff dihubungkan dengan GND dan VCC dari Usb FT232RL. Pin VCC, GND dihubungkan dengan VCC dan GND dari modul AD8232.

Tabel 3.1 Konfigurasi *Port* Mikrokontroler

No	Pin Arduino	Keterangan
1	Pin A0	<i>Output</i> rangkaian LPF
2	Pin 3	Rx
3	Pin 4	Tx
4	Pin 10	<i>Input</i> modul AD8232 (LO+)
5	Pin 11	<i>Input</i> modul AD8232 (LO-)
6	Pin Areff	USB FT232RL
7	Pin VCC, GND	Modul AD8232

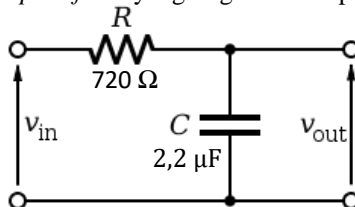


Gambar 3.1 Pin ATMega 328

3.1.2 Rangkaian *Low Pass Filter*

Rangkaian *Low Pass Filter* ini menggunakan rangkaian *low pass filter* yang kapasitif sehingga memakai rangkaian RC. Rangkaian *low pass filter* RC merupakan jenis *filter* pasif, dengan respon frekuensi yang ditentukan oleh konfigurasi R dan C yang digunakan. *Low Pass Filter* adalah *filter* yang berfungsi meneruskan sinyal *input* yang frekuensinya berada dibawah frekuensi tertentu, diatas frekuensi tersebut (frekuensi *cut-off*) sinyal diredam.

Rangkaian *low pass filter* yang di gunakan seperti Gambar 3.2 :



Gambar 3.2 Rangkaian *Low Pass Filter*

Nilai dari Frekuensi *cut off* ditentukan 100 Hz. Nilai kapasitor juga ditentukan nilainya terlebih dahulu daripada resistor. Nilai kapasitor ditentukan 0,1 μF maka perhitungannya adalah sebagai berikut.

$$100 = \frac{1}{2\pi RC}$$

$$100 = \frac{1}{2 \times (3,14) \times R \times 2,2 \times 10^{-6}}$$

$$100 = \frac{1 \times 10^6}{6,28 \times R \times 2,2}$$

$$R = \frac{1000000}{100 \times 6,28 \times 2,2}$$

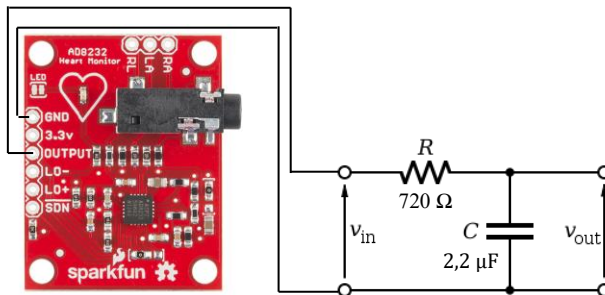
$$R = \frac{1000000}{1381,6}$$

$$R = 723,8 \Omega$$

Karena nilai resistor dari perhitungannya tidak ada di pasaran maka resistor yang digunakan adalah resistor 500 Ω dan 220 Ω . Jadi komponen yang dibutuhkan dalam membuat rangkaian *low pass filter* pada alat ini adalah resistor sebesar 720 Ω dan kapasitor sebesar 2,2 μF .

3.1.3 Perancangan Modul AD8232 Dihubungkan Rangkaian LPF

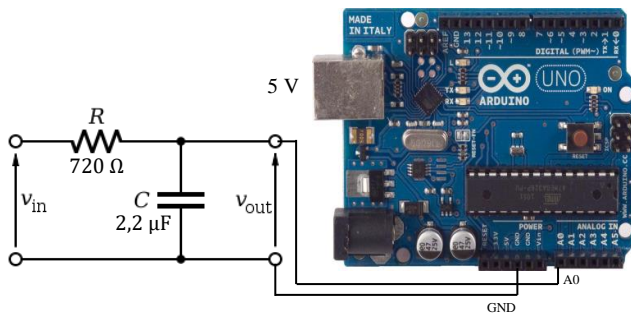
Berdasarkan Gambar 3.3, pada perancangan ini *output* dari AD8232 dihubungkan dengan input (+) pada rangkaian *low pass filter*. Kemudian GND pada AD8232 dihubungkan pada GND rangkaian LPF. *Output* dari rangkaian LPF dihubungkan pada mikrokontroler Arduino UNO.



Gambar 3.3 Konfigurasi AD8232 ke Rangkaian LPF

3.1.4 Perancangan Rangkaian LPF Dihubungkan Mikrokontroler

Berdasarkan Gambar 3.4, pada perancangan ini *output* (+) pada rangkaian *low pass filter* dihubungkan dengan pin A0 pada mikrokontroler Arduino UNO. Kemudian *output* (-) dihubungkan dengan pin GND pada Arduino UNO.



Gambar 3.4 Konfigurasi Rangkaian LPF ke Mikrokontroler

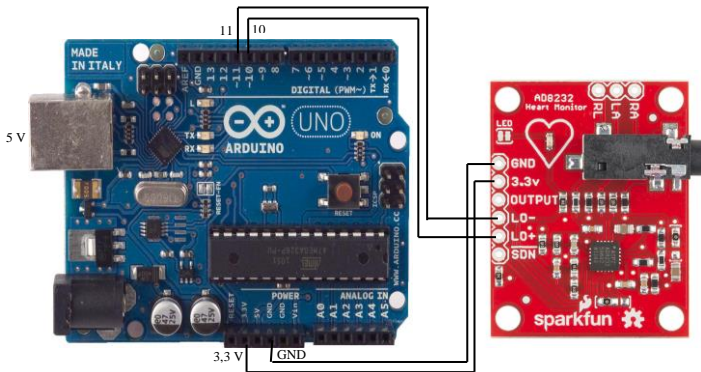
3.1.5 Perancangan Modul AD8232 Dihubungkan Mikrokontroler

Modul AD8232 memiliki beberapa pin *out* yaitu VCC 3,3 V, GND, LO+, LO-, *Output* dan SND. Pin-pin tersebut dihubungkan dengan mikrokontroler arduino UNO sehingga hasil sinyal ECG bisa terbaca dengan program arduino. Modul ini mendapatkan sumber dari arduino melalui pin VCC 3,3 V.

Berdasarkan Gambar 3.5, pin GND pada *board* dihubungkan dengan GND pada arduino. Kemudian pin 3,3 V dihubungkan dengan 3,3 V pada arduino. Pin *output* pada *board* ECG tidak langsung dihubungkan ke pin A0 arduino tapi di *filter* terlebih dahulu dengan rangkaian *low pass filter* kemudian dari *low pass filter* dihubungkan ke A0 dari arduino. Kemudian pin *board* LO- dihubungkan dengan pin 11 pada arduino dan LO+ dihubungkan dengan pin 10 arduino. Pin SDN pada modul AD8232 tidak digunakan karena pin ini hanya bersifat *shutdown*. Konfigurasi pin Arduino ke AD8232 seperti Tabel 3.2

Tabel 3.2 Konfigurasi Arduino ke AD8232

<i>Board Label</i>	<i>Pin Function</i>	<i>Arduino Connection</i>
GND	<i>Ground</i>	GND
3,3 V	<i>3,3 V Power supply</i>	3,3 V
<i>Output</i>	<i>Output Signal</i>	Rangkaian LPF
LO-	<i>Leads-off Detect -</i>	11
LO+	<i>Leads-off Detect +</i>	10
SDN	<i>Shutdown</i>	Tidak digunakan



Gambar 3.5 Konfigurasi Arduino ke AD8232

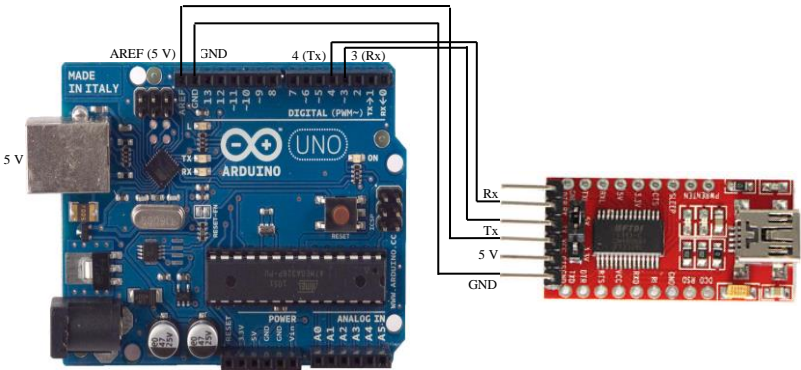
3.1.6 Perancangan FT232RL Dihubungkan Arduino UNO

USB FT232RL adalah modul konversi sinyal USB ke sinyal TTL/UART (*USB-to-TTL Converter*) yang andal dan praktis untuk digunakan pada rangkaian elektronika berbasis mikrokontroler. Modul FT232 berfungsi untuk merubah data USB yang berasal dari port USB menjadi data serial dengan level tegangan TTL. Sehingga dengan menggunakan modul FT232 ini komunikasi data serial (UART) melalui port USB dapat dilakukan. *Board* ini mempunyai LED TX dan RX yang memungkinkan kita untuk melihat lalu lintas serial data melalui LED tersebut. USB ini mendapatkan sumber dari PC/Laptop.

Berdasarkan Gambar 3.6, Pin GND pada USB FT232RL dihubungkan dengan pin GND pada Arduino UNO. Kemudian pin VCC 5 V dihubungkan dengan pin Areff pada Arduino UNO. Setelah itu Rx pada USB di hubungkan dengan Tx pada arduino yang mana tidak mengambil pin Tx (1) dari arduino, akan tetapi memanfaatkan pin 4 sebagai Tx pada arduino. Selanjutnya pin Tx pada USB dihubungkan dengan pin Rx pada arduino yang mana tidak mengambil pin Rx (0), akan tetapi mengkonfigurasi pin 3 arduino menjadi pin Rx. Pin DTR dan CTS pada USB tidak digunakan sehingga tidak dihubungkan dengan Arduino. Konfigurasi pin USB FT232RL ke Arduino UNO seperti Tabel 3.3.

Tabel 3.3 Konfigurasi Pin USB FT232RL ke Pin Arduino UNO

Pin USB	Pin Function	Arduino Connection
GND	Ground	GND
VCC	3,3 V/5 V Power supply	AREF
Rx	Jalur data masuk	Tx (pin 4)
Tx	Jalur data keluar	Rx (pin 3)
CTS	Jalur masuk sinyal <i>clear to send</i>	Tidak digunakan
DTR	Jalur keluar data terminal <i>ready</i>	Tidak digunakan



Gambar 3.6 Konfigurasi Arduino ke USB TTL FT232RL

3.2 Perancangan Software

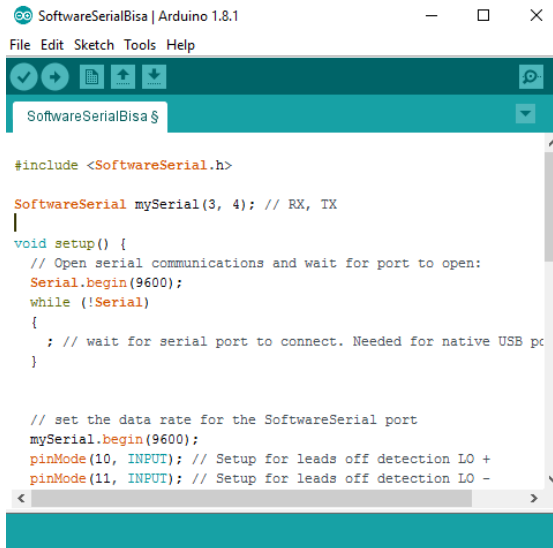
Pada sub bab ini akan dibahas mengenai program pembacaan sensor ECG modul AD8232 pada Mikrokontroler, program untuk menghasilkan nilai data *integer* dari Mikrokontroler, program pembacaan sensor ECG yang ditampilkan pada Delphi berupa hasil sinyal dan nilai *setting point* dengan menerjemahkan nilai *integer* dari Mikrokontroler yang telah terhubung dengan ECG 8232 dan *low pass filter* serta sinyal dari pengambilan data hasil sensor ECG pada Delphi. Hasil yang ditampilkan nanti berupa sinyal ECG, sinyal LPF, sinyal

squaring, sinyal *derivative*, sinyal *moving window*. Harapannya dari tampilan tersebut, ketika dilakukan pengukuran ECG menjadi lebih akurat dan mempermudah dalam mendeteksi Arithmia.

3.2.1 Program Pembacaan Modul AD8232 Mikrokontroler

Pembacaan ini berguna untuk menampilkan *output* sinyal analog ECG yang ditampilkan oleh serial *plotter* dari *software* IDE Arduino. Program ini membaca nilai analog dari modul AD8232 dengan fungsi *mySerial.println* (*analogRead*(A0)) . pada fungsi ini menggunakan *mySerial* karena USB yang digunakan bukan serial COM arduino melainkan USB TTL FT232RL.

Program pada *software* IDE arduino untuk menampilkan hasil sinyal pada serial *plotter software* arduino seperti Gambar 3.7 dan 3.8. Program perintah untuk membaca *output* analog seperti Gambar 3.9.



```
SoftwareSerialBisa | Arduino 1.8.1
File Edit Sketch Tools Help

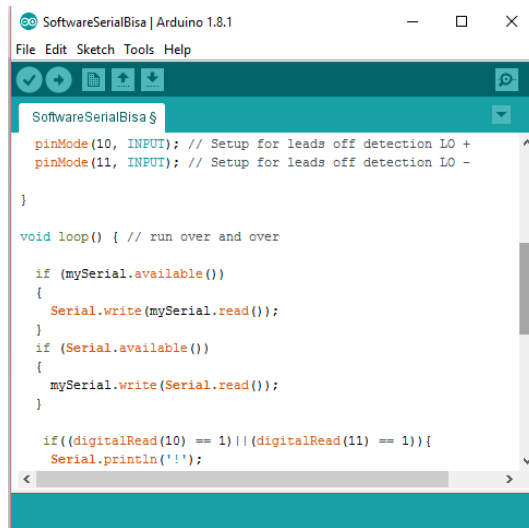
#include <SoftwareSerial.h>

SoftwareSerial mySerial(3, 4); // RX, TX

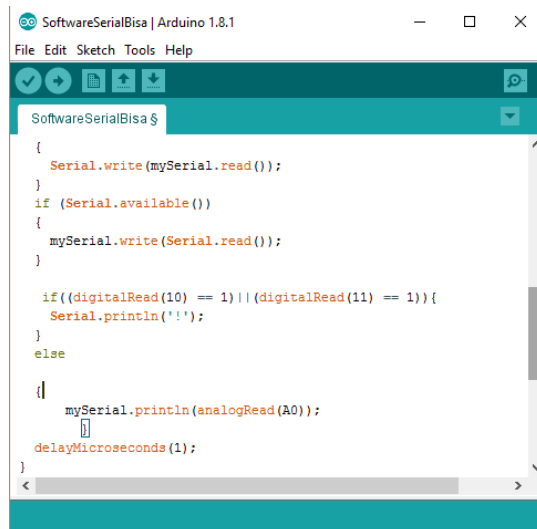
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial)
  {
    ; // wait for serial port to connect. Needed for native USB port
  }

  // set the data rate for the SoftwareSerial port
  mySerial.begin(9600);
  pinMode(10, INPUT); // Setup for leads off detection LO +
  pinMode(11, INPUT); // Setup for leads off detection LO -
}
```

Gambar 3.7 Program Arduino untuk Menampilkan Sinyal ECG AD8232 pada Serial *Plotter*



Gambar 3.8 Lanjutan Program Arduino untuk Menampilkan Sinyal ECG AD8232 pada Serial *Plotter*



Gambar 3.9 Perintah Membaca *Output* Analog pada AD8232

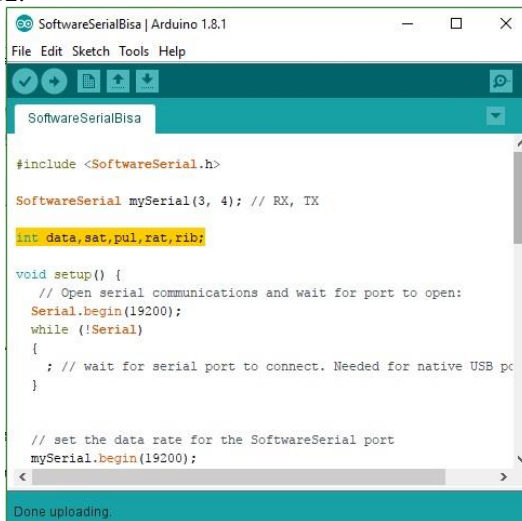
3.2.2 Program Pembacaan Nilai Data *Integer* pada Mikrokontroler

Program ini untuk membaca nilai data *integer* yang nantinya diterjemahkan pada *software* Delphi untuk menampilkan *software* QRS *Detector*. Hasil dari nilai data ini bisa dilihat pada serial monitor pada arduino dengan serial com yang digunakan untuk USB FT232RL karena serial yang digunakan adalah USB FT232RL. Pada program ini nilai data yang dihasilkan adalah nilai 4 digit data interger yang diterjemahkan sebagai berikut:

- int rib : yaitu untuk menampilkan nilai data ribuan
- int rat : yaitu untuk menampilkan nilai data ratusan
- int pul : yaitu untuk menampilkan nilai data puluhan
- int sat : yaitu untuk menampilkan nilai data satuan.

Pada program ini pembacaan menggunakan fungsi *mySerial* karena connector yang dihubungkan pada PC bukan serial com biasa melainkan USB TTL FT232RL. *Delay* yang digunakan pada program ini adalah 1 mikrosekon.

Program pada *software* IDE arduino untuk pembacaan nilai data *integer* pada mikrokontroler arduino UNO seperti Gambar 3.10 dan 3.11. Program perintah untuk membaca nilai data *integer* seperti Gambar 3.12.



```
SoftwareSerialBisa | Arduino 1.8.1
File Edit Sketch Tools Help

SoftwareSerialBisa

#include <SoftwareSerial.h>

SoftwareSerial mySerial(3, 4); // RX, TX

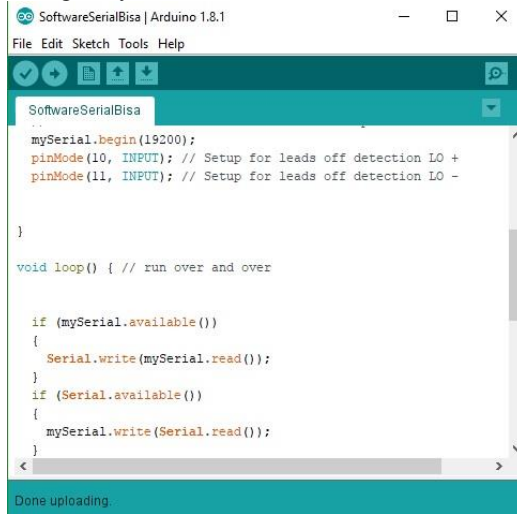
int data,sat,pul,rat,rib;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(19200);
  while (!Serial)
  {
    ; // wait for serial port to connect. Needed for native USB port
  }

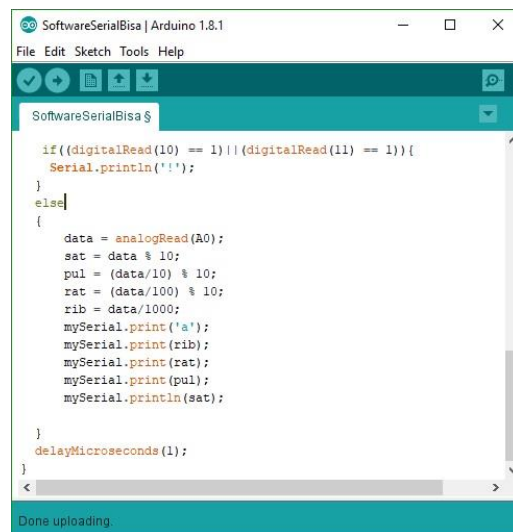
  // set the data rate for the SoftwareSerial port
  mySerial.begin(19200);
}
```

Gambar 3.10 Program Pembacaan Nilai Data *Integer*

Rx dan Tx arduino menggunakan pin 3 dan 4 sehingga Rx dan Tx arduino diterjemahkan pada *software* IDE arduino sebagai *software* serial dengan fungsi mySerial (3, 4).



Gambar 3.11 Lanjutan Program Pembacaan Nilai Data *Integer*



Gambar 3.12 Perintah Pembacaan Nilai Data *Integer*

3.2.3 Perancangan Software QRS Detector

Program pembacaan sensor ECG yang ditampilkan pada Delphi berupa tampilan sinyal ECG seperti Gambar 3.14. Program nilai *setting point* dengan menerjemahkan nilai *integer* Gambar 3.13. Hasil yang ditampilkan nanti juga berupa sinyal ECG, sinyal LPF (*low pass filter*), sinyal HPF (*high pass filter*), sinyal *squaring*, sinyal *derivative*, sinyal *moving window*. Hasil sinyal tersebut didapatkan dengan memasukkan fungsi persamaan metode QRS Detector ke *code editor software delphi* menjadi suatu program,

Program untuk mendapatkan sinyal ECG seperti Gambar 3.15. Kemudian program untuk mendapatkan sinyal LPF seperti Gambar 3.17. Setelah itu program untuk mendapatkan sinyal HPF seperti Gambar 3.16. Program untuk mendapatkan sinyal *derivative* seperti Gambar 3.18. Kemudian program untuk mendapatkan sinyal *squaring* seperti Gambar 3.19. Setelah itu untuk mendapatkan sinyal *moving* menggunakan program seperti Gambar 3.20

Pada tampilan delphi, ditampilkan sinyal ECG dari hasil *interface* antara delphi dengan arduino yang telah dihubungkan dengan modul AD8232 dan rangkaian *low pass filter*. Kemudian nilai *setting point* dan tegangan juga ada pada tampilan delphi. Setelah itu ditampilkan beberapa proses dari sinyal ECG mulai dari hasil sinyal ECG yang di *filter* dengan menggunakan fungsi *low pass filter* menjadi sinyal LPF. Kemudian di *filter* menggunakan persamaan *high pass filter* menjadi sinyal HPF. Setelah itu dilakukan diferensiasi untuk mencari kemiringan / lereng kompleks QRS sehingga menjadi sinyal *derivative*. Selanjutnya dilakukan pengkuadratan untuk mendapatkan sinyal *squaring* dan memasukkan persamaan *moving window* untuk mendapatkan sinyal *moving window*.

Tampilan dari *software delphi* yang diberi nama *software QRS Detector* seperti Gambar 3.21.

```
rib := StrToIntDef(rib1,0);  
sumbuX[i] := rib;  
volt1:= rib * (5/1023);  
Edit4.Text := FloatToStr(volt1);  
Edit1.Text := IntToStr(sumbuX[i]);
```

Gambar 3.13 Program Pembacaan Nilai *Setting Point*

```

        i:=i+1;

        if i=400 then
        begin
            Series1.Clear;

            for i:=0 to 400 do
            begin
                Series1.AddXY(i,sumbuX[i],'',clRed);
            end;
        end;
    end;
end;

```

Gambar 3.14 Program Pembacaan Tampilan ECG 400 Data

```

procedure TForm1.EKG;
begin
    for t:=1 to 200 do
    begin
        ecg[t]:=0;
        end;
        for t:=1 to 200 do
        begin
            ecg[t]:=sumbuX[t];
        end;
    end;
end;

```

Gambar 3.15 Program Pembacaan Sinyal ECG

```

procedure TForm1.HPF;
begin
    for t:= 1 to 200 do
    begin
        if t = 1 then y2[t] :=-y1[t]/32
        else
            if t<=16 then y2[t] :=-(y2[t-1]+y1[t])/32
            else
                if t<=32 then y2[t] := y1[t-16]-(y2[t-1]+y1[t])/32
                else
                    y2[t]:=y1[t-16]-(y2[t-1]+y1[t]-y1[t-32])/32;
            end;
        end;
    end;
end;

```

Gambar 3.16 Program Pembacaan Sinyal HPF

```

procedure TForm1.LPF;
begin
  for t:= 1 to 200 do
    begin
      if t = 1 then y1[t]:= ecg[t]
      else
        if t = 2 then y1[t]:= 2*y1[t-1]+ ecg[t]
        else
          if t<= 6 then y1[t]:= 2*y1[t-1]-y1[t-2]+ ecg[t]
          else
            if t<= 12 then y1[t]:=2*y1[t-1]-y1[t-2]+ ecg[t]-2*ecg[t-6]
            else
              y1[t]:=2*y1[t-1]-y1[t-2]+ ecg[t]-2*ecg[t-6]+ ecg[t-12];
    end;
  for t:=0 to 200 do
    begin
      Series4.Addxy(t,y1[t],'',clred);
      delay(1);
    end;
end;

```

Gambar 3.17 Program Pembacaan Sinyal LPF

```

Procedure TForm1.Derivative;
begin
  for t :=1 to 200 do
    begin
      if t=1 then y3[t]:=2*y2[t]/8
      else
        if t<=3 then y3[t]:=(2*y2[t]+y2[t-1])/8
        else
          if t<=4 then y3[t]:=(2*y2[t]+y2[t-1]-y2[t-3])/8
          else
            y3[t]:=(2*y2[t]+ y2[t-1]-y2[t-3]-2*y2[t-4])/8;
    end;
  for t:=0 to 200 do
    begin
      Series6.Addxy(t,y3[t],'',clred);
      delay(1);
    end;
end;

```

Gambar 3.18 Program Pembacaan Sinyal *Derivative*

```

procedure TForm1.Squaring;
begin
  for t:=1 to 200 do
    y4[t]:=y3[t] * y3[t];
  for t:=0 to 200 do
    begin
      Series7.Addxy(t,y4[t],'',clred);
      delay(1);
    end;
end;

```

Gambar 3.19 Program Pembacaan Sinyal *Squaring*

```

procedure TForm1.Moving;
begin
  for t :=1 to 200 do
    begin
      if t =1 then y5[t] :=y4[t]/16
      else
        if t <=8 then
          begin
            temp:=0;
            for m :=1 to t-1 do
              begin
                temp:=temp+y4[t-m];
              end;
            y5[t]:=(y4[t]+temp)/16;
          end
        else
          temp:=0;
          for m:=1 to 7 do
            begin
              temp:=temp+y4[t-m];
            end;
            y5[t]:=(y4[t]+temp)/16;
          end;
        for t:=0 to 200 do
          begin
            Series8.Addxy(t,y5[t],'',clred);
            delay(1);
          end;
        end;
      end;
end;

```

Gambar 3.20 Program Pembacaan Sinyal *Moving Window*



Gambar 3.21 Tampilan *Software QRS Detector* pada Delphi

BAB IV

PENGUKURAN DAN ANALISA

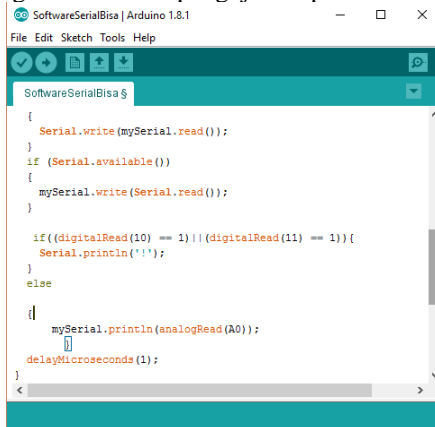
Pengujian pada sistem instrument detektor arithmia ini meliputi: mencari *heart rate* dan pengujian keseluruhan. Setelah melakukan beberapa pengujian alat, data yang diperoleh dianalisa untuk mengetahui proses kerja dari seluruh sistem instrument detektor arithmia.

4.1 Pengujian Sensor Elektroda dan Board AD8232

Pengujian ini menggunakan sensor elektroda yang terbuat dari Ag/AgCl yang dapat ditempel ke bagian tubuh. Kemudian probe dari *lead* elektroda di dihubungkan dengan *board* AD8232 yang terhubung dengan Arduino . Pada pengujian ini ada 2 cara yaitu :

1. *Lead* berwarna merah yang bertuliskan (R) di tempelkan pada lengan sebelah kanan, kemudian *lead* berwarna kuning yang bertuliskan (L) ditempelkan pada lengan sebelah kiri dan *lead* berwarna hijau yang bertuliskan (F) ditempelkan pada kaki sebelah kiri.
2. *Lead* berwarna merah yang bertuliskan (R) di tempelkan pada dada sebelah kanan, kemudian *lead* berwarna kuning yang bertuliskan (L) ditempelkan pada dada sebelah kiri dan *lead* berwarna hijau yang bertuliskan (F) ditempelkan dibawah ulu hati agak ke sebelah kanan.

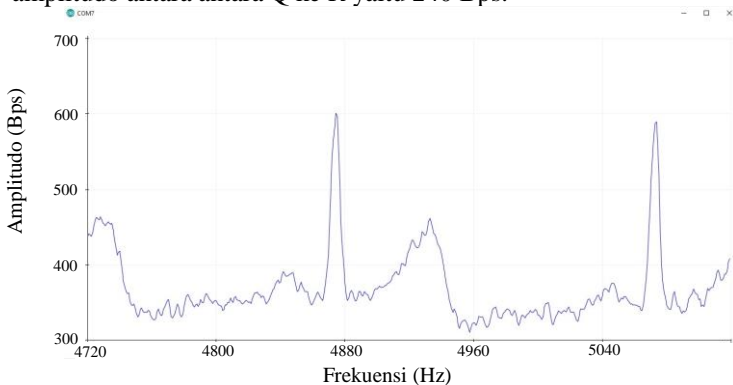
Program yang digunakan untuk pengujian seperti Gambar 4.1.



Gambar 4.1 Program Pengujian Elektroda dan AD8232

4.1.1 Hasil Uji Elektroda dan AD8232 Cara 1

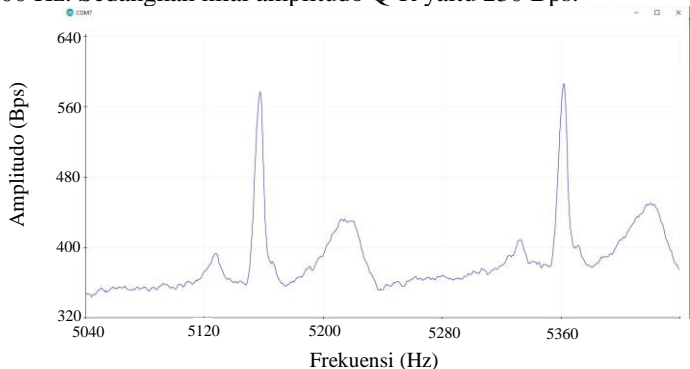
Pada pengujian ini hasil ditampilkan pada serial monitor *software* IDE. Berdasarkan Gambar 4.2, hasil dari pengujian ini yaitu setiap 1 sinyal ECG memiliki jarak Puncak R ke R (R-R) yaitu 200 Hz. nilai amplitudo antara antara Q ke R yaitu 240 Bps.



Gambar 4.2 Hasil Uji Elektroda Cara 1

4.1.2 Hasil Uji Elektroda dan AD8232 Cara 2

Pada pengujian ini hasil ditampilkan pada serial monitor *software* IDE. Berdasarkan Gambar 4.3, hasil dari pengujian ini yaitu setiap 1 sinyal ECG kompleks QRS memiliki jarak puncak R ke R (R-R) yaitu 200 Hz. Sedangkan nilai amplitudo Q-R yaitu 230 Bps.



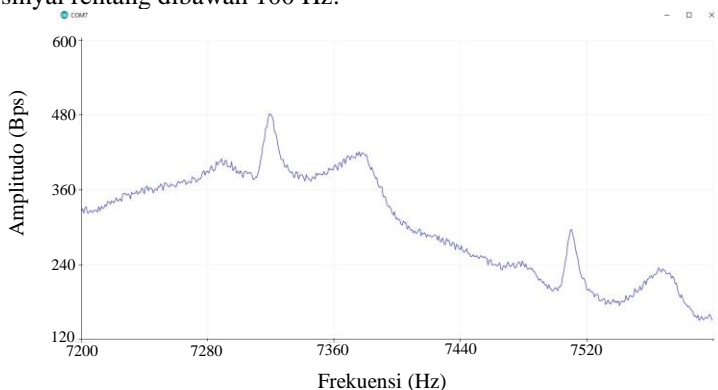
Gambar 4.3 Hasil Uji Elektroda Cara 2

4.2 Pengujian Hasil AD8232 yang Dihubungkan Rangkaian *Low Pass Filter*

Pada pengujian ini program yang digunakan menggunakan program pengujian elektroda dan AD8232 seperti Gambar 4.1. Hanya saja *output* dari *board* AD8232 dihubungkan pada *input* rangkaian *low pass filter* resistif kapasitif (RC). Rangkaian *low pass filter* ini. Hasil pada pengujian ini bisa mengurangi sinyal–sinyal interferensi yang dikarenakan interferensi dari pergerakan atau sinyal otot. Pengujian ini juga di ambil dengan 2 cara.

4.2.1 Hasil Uji Sinyal ECG AD8232 Dihubungkan Rangkaian *Low Pass Filter* Cara 1

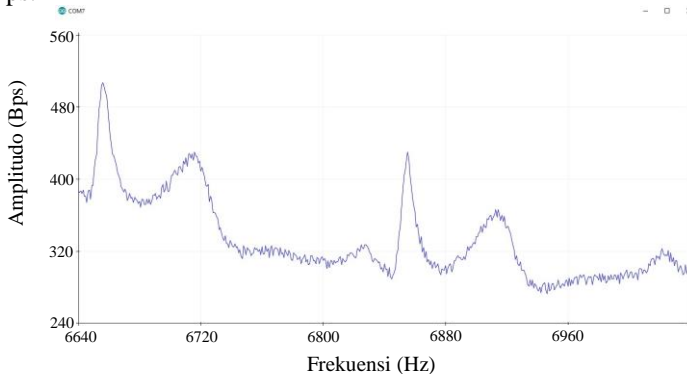
Pada pengujian ini menggunakan cara 1 yaitu ditempelkan pada lengan tangan dan kaki. Berdasarkan Gambar 4.4, hasil dari pengujian ini yaitu sinyal ECG yang ditampilkan oleh serial monitor arduino lebih jelas, dikarenakan hasil dari *output* AD8232 di hubungkan rangkaian *low pass filter* maka hasil dari pengujian ini yaitu setiap 1 sinyal ECG komplek QRS memiliki jarak puncak R ke R (R–R) yaitu 200 Hz. Sedangkan nilai amplitudo Q–R yaitu 100–120 Bps. Melihat hasil dari pengujian ini, rangkaian *low pass filter* berfungsi dengan meloloskan sinyal rentang dibawah 100 Hz.



Gambar 4.4 Hasil Uji AD8232 Dihubungkan Rangkaian LPF Cara 1

4.2.2 Hasil Uji Sinyal ECG AD8232 Dihubungkan Rangkaian *Low Pass Filter* Cara 2

Pada pengujian ini menggunakan cara 2 yaitu ditempelkan pada dada dan perut agak ke kanan mendekati ulu hati. Berdasarkan Gambar 4.5, hasil dari pengujian ini yaitu sinyal ECG yang ditampilkan oleh serial monitor arduino lebih jelas, dikarenakan hasil dari *output* AD8232 di hubungkan rangkaian *low pass filter* maka hasil dari pengujian ini yaitusetiap 1 sinyal ECG kompleks QRS memiliki jarak puncak R ke R (R–R) sebesar 200 Hz. Sedangkan nilai amplitudo Q–R yaitu 100 – 120 Bps.



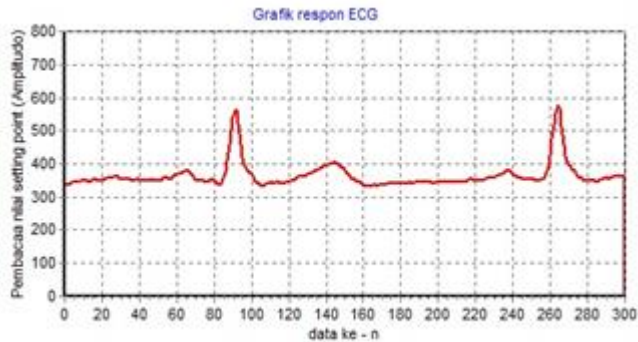
Gambar 4.5 Hasil Uji AD8232 Dihubungkan Rangkaian LPF Cara 2

4.3 Pengujian Rangkaian *Low Pass Filter*

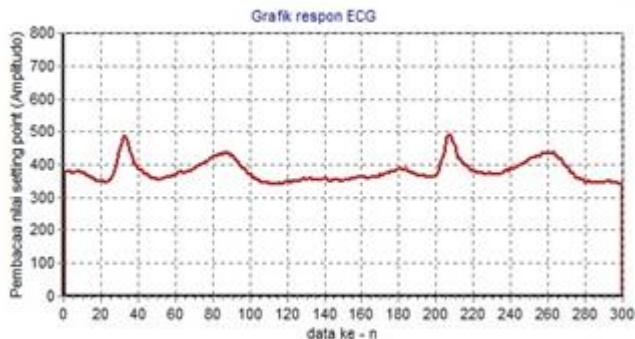
Pada pengujian ini dilakukan perbandingan dengan membandingkan hasil sinyal ECG ketika tanpa dihubungkan rangkaian *low pass filter* dan ketika dihubungkan rangkaian *low pass filter*. Hasil sinyal ECG ditampilkan pada tampilan respon ECG *software* QRS *detector*. Program yang digunakan untuk pengujian ini seperti Gambar 3.14.

Berdasarkan Gambar 4.6, ketika *output* AD8232 tidak dihubungkan dengan rangkaian *low pass filter* maka hasil pada nilai amplitudo yang melebihi frekuensi 100 Hz tetap diloloskan. Pada tampilan hasil tampilan respon ECG nilai amplitudo puncak R yaitu mencapai frekuensi 250 Hz. Berdasarkan Gambar 4.7 ketika *output* AD8232 dihubungkan dengan rangkaian *low pass filter* maka hasil pada nilai amplitudo yang melebihi frekuensi 100 Hz di hilangkan, tetapi sinyal pada frekuensi amplitudo dibawah 100 Hz tetap diloloskan. Pada

tampilan respon ECG ketika dihubungkan rangkaian *low pass filter* nilai amplitudo puncak R yaitu berkisar 100–120 Hz.



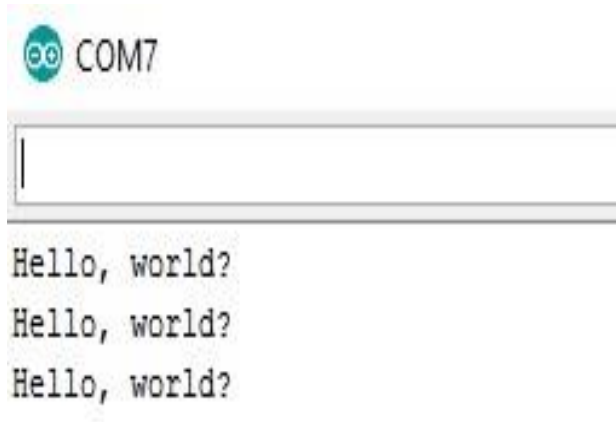
Gambar 4.6 Hasil Tampilan ECG Sebelum Dihubungkan Rangkaian *Low Pass Filter*



Gambar 4.7 Hasil Tampilan ECG Setelah Dihubungkan Rangkaian *Low Pass Filter*

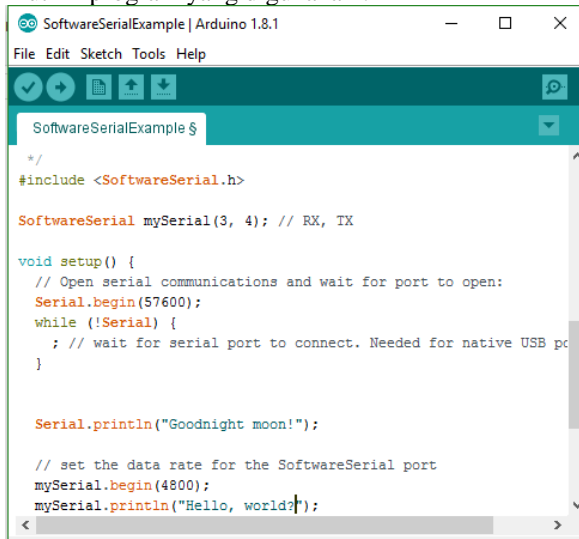
4.4 Pengujian Pengiriman Data Oleh USB FT232RL

Pada pengujian ini yang dilakukan yaitu dengan membuat program untuk mengirim data melalui USB FT232 RL yang ditampilkan di serial monitor *software* IDE seperti pada Gambar 4.9 dan Gambar 4.10 . Pengujian yang dilakukan dengan mengirimkan kalimat “*Hello World*” yang ditampilkan pada serial monitor *software* IDE seperti pada Gambar 4.8.

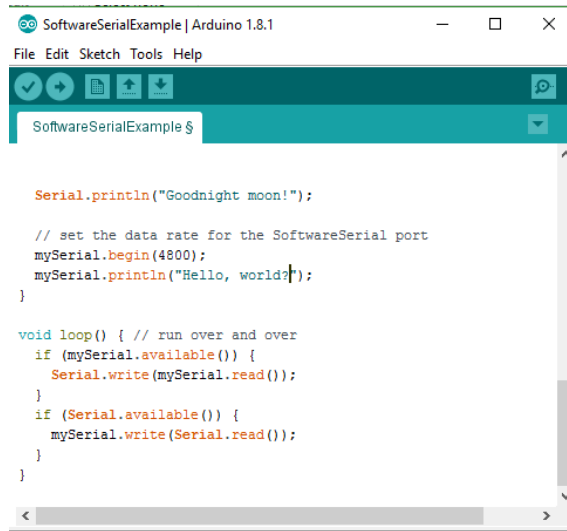


Gambar 4.8 Hasil Uji USB FT232 RL

Berikut ini program yang digunakan :



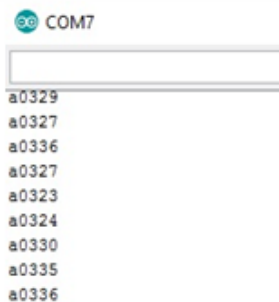
Gambar 4.9 Program Pengujian USB



Gambar 4.10 Lanjutan Program Pengujian USB

4.5 Pengujian Pembacaan Data *Integer* Arduino yang Dihubungkan Modul AD8232

Pada pengujian ini, dilakukan pembacaan data *integer* dengan mencantumkan 4 digit angka. 4 digit angka tersebut terdiri dari ribuan, ratusan, puluhan dan satuan. Kemudian data yang didapat dihitung. Program yang digunakan untuk mengetahui nilai data integer seperti Gambar 3.12. Berdasarkan Gambar 4.11, data yang diperoleh yaitu berkisar antara 295–450.



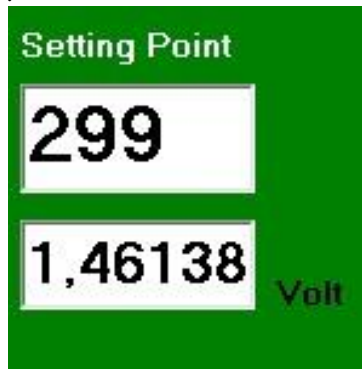
Gambar 4.11 Hasil Pembacaan Data *Integer*

4.6 Pengujian Instrument Detektor Arithmia dengan *Software QRS Detector*

Pada pengujian ini dilakukan pada seorang remaja berusia 20 Th. Caranya dengan melakukan *interface* antara arduino ke *software delphi* untuk menerjemahkan agar bisa dibaca oleh delphi maka pada delphi dibuat program untuk membaca data *integer* yang telah di program pada arduino untuk menampilkan nilai tegangan dan nilai data integer seperti Gambar 3.13. Kemudian hasil data di tampilkan dalam bentuk sinyal menjadi sinyal ECG dengan menggunakan program seperti Gambar 3.14. *Setting point* (data *integer*) ini di simpan hingga 400 data. Setelah itu di tampilkan dalam bentuk tampilan respon ECG pada *form*. Selanjutnya ditampilkan sinyal ECG, sinyal LPF, sinyal HPF, sinyal *derivative*, sinyal *squaring* dan sinyal *moving*.

4.6.1 Hasil Pembacaan *Setting Point*

Hasil yang ditampilkan pada *form delphi* yaitu berupa *setting point* yang berjalan hingga mencapai 400 data. Kemudian *setting point* berhenti dan menampilkan hasil sinyal ECG dengan data yang telah tersimpan. Berdasarkan Gambar 4.12, hasil nilai data yaitu berkisar antara 295 – 450. Kemudian untuk nilai tegangan yaitu berkisar antara 1,3 V sampai 2,0 V.



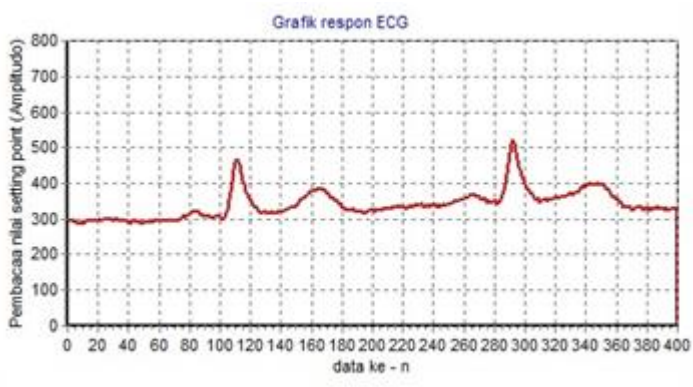
Gambar 4.12 Hasil Tampilan Pembacaan *Setting Point*

4.6.2 Hasil Tampilan Respon ECG

Hasil yang ditampilkan pada *form delphi* yaitu berupa sinyal ECG. Tampilan sinyal ini mengambil data dari penyimpanan *setting point* hingga 400 data. Dari 400 data tersebut ditampilkan dalam bentuk sinyal

oleh delphi sehingga membentuk sinyal ECG . Berdasarkan Gambar 4.13, jarak antara puncak R–R yaitu 180 Hz sehingga Dalam jarak 400 Hz didapatkan 2–3 sinyal ECG komplek QRS sehingga nilai *heart rate* pada tampilan ini yaitu sekitar 60–70 Bpm. Pada saat *chart* membaca hasil ECG, *setting point* berhenti kemudian merekam kembali hingga 400 data.

Berikut ini tampilan respon ECG :



Gambar 4.13 Hasil Tampilan Respon ECG

4.7 Signal Processing

Pada pengujian ini yang dilakukan adalah memasukkan persamaan metode QRS *detector* dalam program delphi. Hasil dari respon ECG yang ditampilkan oleh *form* tampilan respon ECG di bagi menjadi 6 sinyal yang digunakan untuk menganalisa sinyal ECG sehingga pemeriksa dapat dengan mudah mengetahui hasil pemeriksaan pasien dengan baik. Keenam sinyal tersebut diambil 200 data saja dan didapatkan dari hitungan persamaan pada program di *software* delphi. Pembagian sinyal tersebut yaitu

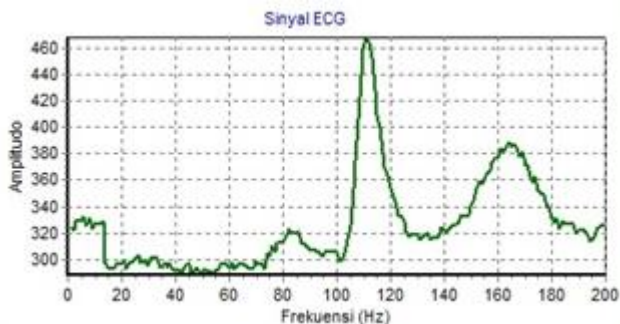
- A. Sinyal ECG
- B. Sinyal LPF ECG
- C. Sinyal HPF ECG
- D. Sinyal *Squaring*
- E. Sinyal *Derivative*
- F. Sinyal *Moving*

Keenam sinyal tersebut menjadi landasan untuk menghasilkan hasil pemeriksaan yang akurat.

4.7.1 Hasil Sinyal ECG

Pada hasil sinyal ECG ini didapatkan sama seperti pembacaan tampilan respon ECG. Proses untuk menampilkan sinyal ECG yaitu menggunakan program seperti Gambar 3.15. Hasil yang ditampilkan pada tampilan sinyal ECG tidak berbeda dengan tampilan respon ECG, hanya saja pada tampilan ini diambil 200 data sehingga tampilan lebih jelas. Hasil sinyal ECG ini terdapat kompleks QRS. pada analisa sinyal ECG ini yang di tinjau adalah hasil puncak R–R . Berdasarkan Gambar 4.14, sinyal ECG yang didapat pada pengujian terdapat 1–2 sinyal ECG setiap frekuensi 200 Hz dalam 1 detik sehingga *heart rate* yang didapat per menit yang didapat mencapai 60–70 bpm. Nilai amplitudo pada sinyal ini berkisar antara 1,4 V – 2,2 V. nilai amplitudo puncak R yaitu $470 \times 5 : 1023 = 2,2$ V.

Berikut ini gambar tampilan hasil sinyal ECG:



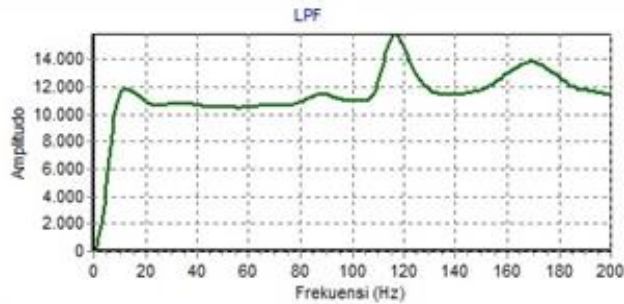
Gambar 4.14 Tampilan Sinyal ECG

4.7.2 Hasil Sinyal LPF ECG

Pada pengujian ini menggunakan program seperti pada Gambar 3.17. Frekuensi *cut off* pada *low pass filter* ini yaitu 11 Hz maka terjadi *delay* 5-11 data (25 ms). Setiap kebisingan 60 Hz atau suara bising otot juga akan dilemahkan secara signifikan. Pada persamaan untuk menyaring sinyal menggunakan LPF, nilai frekuensi 60 Hz. Berdasarkan Gambar 4.15, Gain persamaan ini yaitu 36dB. Puncak R berada pada amplitudo bernilai 15000. Maka hasil nilai amplitudo puncak R yaitu $(15000 : 32) \times 5 : 1023 = 2,03$ V . Dari hasil ini maka hasil *filter* terjadi peredaman *noise* yang awalnya nilai amplitudo dari R yaitu berkisar 2,2

V menjadi 2,03 V. Amplitudo pada hasil tampilan ini berkisar antara 1,4 V – 2,03 V.

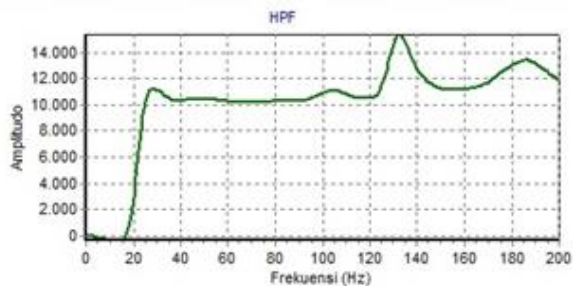
Hasil sinyal LPF ECG pada tampilan QRS *Detector* adalah seperti Gambar 4.15:



Gambar 4.15 Tampilan Sinyal LPF ECG

4.7.3 Hasil Sinyal HPF ECG

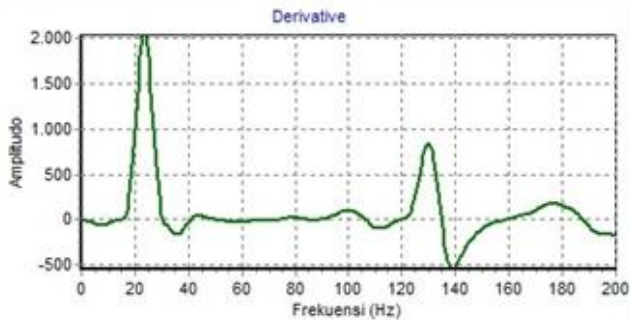
Pada pengujian ini menggunakan program seperti pada Gambar 3.16. *High pass* diperoleh dengan membagi *output* dari *filter* low-pass dengan gain dc *low pass filter* yaitu 32 dB dan kemudian mengurangkannya dari sinyal aslinya. Berdasarkan Gambar 4.16, frekuensi *cut off* pada HPF ini yaitu 5 Hz. *Delay* pada saat masuk persamaan ini yaitu 16 T (16 data atau 80 ms), hasil *high pass filter* ini lanjutan daripada *low pass filter*. Gain dari HPF sendiri yaitu 1. Puncak HPF berada pada amplitudo 14500. Maka nilai amplitudo puncak R yaitu $(14500 : 32) \times 5 : 1023 = 2,21$ V. Nilai dari amplitudo pada tampilan ini yaitu berkisar 1,5 V – 2,21 V.



Gambar 4.16 Tampilan Sinyal HPF ECG

4.7.4 Hasil Sinyal *Derivative*

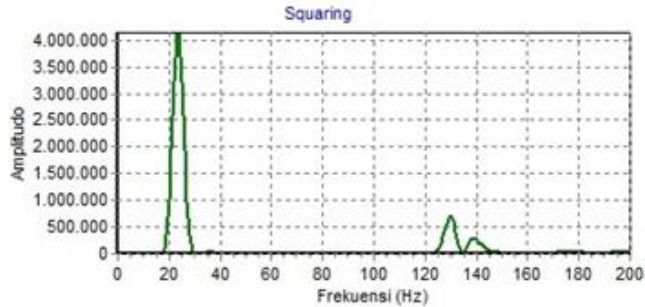
Setelah dilakukan proses pemfilteran, Sinyal didiferensiasi untuk mendapatkan lereng kompleks QRS yang dinamakan sinyal *Derivative*. Sinyal ini merupakan suatu proses turunan dari sinyal yang telah melalui proses pemfilteran pada HPF dan LPF. Program yang digunakan untuk pegujian seperti pada Gambar 3.18. Berdasarkan Gambar 4.17, *delay* pada sinyal ini yaitu $2T$ (10 ms). Memiliki gain 0,1. Nilai frekuensi amplitudo puncak R dari sinyal *derivative* ini yaitu berkisar 800. Frekuensi kompleks QRS yaitu berada pada frekuensi 120 - 140 Hz.



Gambar 4.17 Tampilan Sinyal *Derivative*

4.7.5 Hasil Sinyal *Squaring*

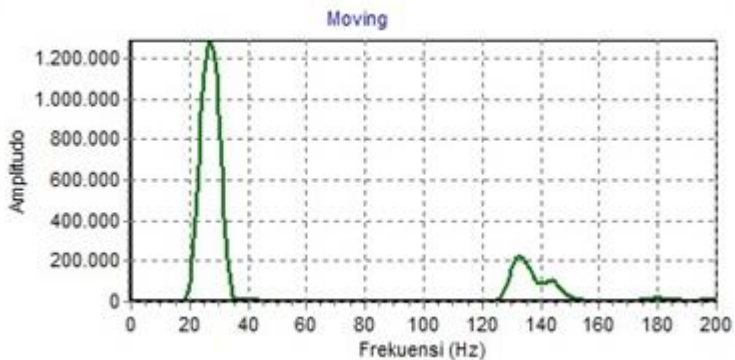
Sinyal *squaring* ini adalah lanjutan dari sinyal *derivative* yang telah di *filter* oleh HPF dan LPF. Cara mendapatkan nilai *squaring* dengan mengkuadratkan nilai sinyal *derivative*. . Program yang digunakan untuk pegujian seperti pada Gambar 3.19. Berdasarkan Gambar 4.18, nilai amplitudo dari sinyal *squaring* ini berkisar 640.000 yaitu hasil kuadrat dari nilai amplitudo puncak R pada sinyal *derivative* yaitu 800 maka nilai amplitudo puncak R pada tampilan ini sama dengan *derivative* pada frekuensi 130 Hz.



Gambar 4.18 Tampilan Sinyal *Squaring*

4.7.6 Hasil Sinyal *Moving*

Sinyal *Moving* selanjutnya digunakan untuk menghaluskan sinyal. Proses ini digunakan untuk mengekstrak fitur QRS. Lebar *window* sama dengan durasi QRS terbesar. . Program yang digunakan untuk pegujian seperti pada Gambar 3.20. Berdasarkan Gambar 4.19, terlihat kejadian perubahan kompleks QRS nya. Amplitudo pada hasil sinyal *moving* yaitu berkisar pada 200.000 terletak pada frekuensi antara 130 Hz. Jarak kompleks QRS berkisar antara 120 - 140 Hz maka hasilnya durasi QRS mencapai 30 *sample* data (150 ms)



Gambar 4.19 Tampilan Sinyal *Moving*

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

5.1 Kesimpulan

Dari perencanaan, pembuatan, pengujian, dan hasil maka dapat disimpulkan sebagai berikut:

1. *QRS Detector* bisa mengurangi interfensi sinyal lain pada rentan gangguan 60 Hz.
2. Rangkaian *low pass filter* pada alat ini dapat memotong amplitudo sinyal pada rentan frekuensi diatas 100 – 130 Hz
3. Data yang didapatkan pada serial monitor setiap detiknya yaitu 200. Sehingga mendapatkan frekuensi *sampling* 200 Hz.
4. Nilai Amplitudo pada hasil sinyal ECG berkisar diantara 1,4 – 2,2 V.
5. Tampilan pada *QRS Detector* yaitu berupa sinyal ECG yang diproses hingga didapatkan sinyal *moving* yang mengilustrasikan kejadian perubahan komplek QRS.
6. Jarak dari puncak R–R yaitu berkisar 180 sehingga pada frekuensi *sampling* yang ditampilkan (200 Hz), mendapatkan 1 - 2 puncak R–R.
7. Nilai *heart rate* pada pengujian instrument detektor arithmia ini ketika keadaan normal yaitu 60–70 bpm.

5.2 Saran

1. Untuk dapat melakukan monitoring ECG dengan jarak yang lebih jauh sebaiknya digunakan media komunikasi WIFI atau bahkan dengan jaringan internet dengan tetap memanfaatkan PC atau *smartphone* Android sebagai pengolah data rekaman ECG.
2. Menambahkan pemberitahuan terdiagnosa atau tidak pada *software* nya sehingga pasien langsung mengerti apakah dia terkena kelainan jantung
3. Diberi indikator pemberitahuan semisal terjadi kesalahan pada alat.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] *Datasheet Microcontroller ATmega 328*.
<http://www.alldatasheet.com>. Diakses tanggal 26 April 2017.
- [2] Kadir, Abdul. *Panduan Mempelajari Arduino*. Jogjakarta. ANDI OFFSET. 2013.
- [3] Tompkins W. J. A. *Biomedical Digital Signal Processing*. Prentice-Hall. New Jersey. 2000.
- [4] *Datasheet AD8232*. www.analog.com. Di akses tanggal 10 Maret 2017.
- [5] Hampton, Jhon R. *Dasar-Dasar EKG*. Jakarta. EGC. 2006.
- [6] Ismail, D. F. Rancang Bangun Portabel Tensimeter dan Elektrokardiograf Berbasis Mikrokontroler. *Buku Tugas Akhir*. Jurusan Fisika. Institut Teknologi Sepuluh Nopember Surabaya . 2017.
- [7] *AD8232 Heart Rate Monitor Hookup Guide*.
<https://learn.sparkfun.com/tutorials/ad8232-heart-rate-monitorhookup-guide/all.pdf>. Diakses tanggal 14 Maret 2017.
- [8] *Data Sheet FT 232R*. <http://www.ftdichip.com/Products/ICs/FT232R.htm>. Diakses tanggal 13 Juni 2017
- [9] Bakri, Abdul Haris, dkk. *Dasar-Dasar Elektronika Buku 1*. Makassar: UNM. 2008.
- [10] Inge Martina. *36 Jam Belajar Komputer Pemrograman Visual Borland Delphi 7*. Semarang. PT Elex Media dan Wahana Komputer. 2004.
- [11] *Anatomi & Fisiologi Jantung*. <https://bukusakudokter.org/2012/10/14/anatomi-fisiologi-jantung/amp/>. Diakses tanggal 13 Juni 2017
- [12] Sudoyo. *Ilmu Penyakit Dalam Edisi V*. Jakarta. Interna Publishing. 2009.
- [13] Uuvuori, Johana. *Arrhythmia Analysis (Heart Rate Variability)*. www.cis.hut.fi/Opinnot/T-61.6010/s04/local/uusvuori.ppt. Diakses tanggal 11 Juni 2017.
- [14] Lauralee, Sherwood. *Fisiologi Manusia dari Sel ke Sistem*. Jakarta. EGC. 2001.


-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

DATASHEET

1. Datasheet Arduino UNO

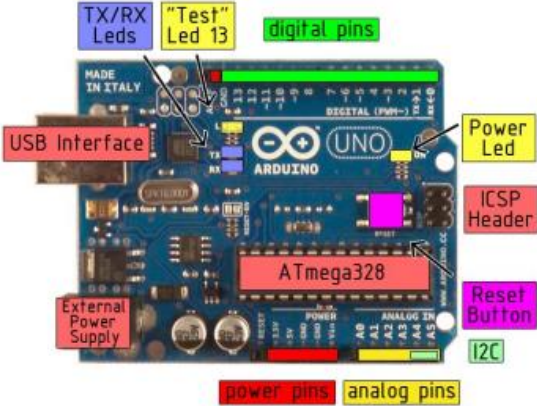
Technical Specification







EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary	
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In Tools>Board select

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



Done compiling

Press Compile button
(to check for errors)



Upload



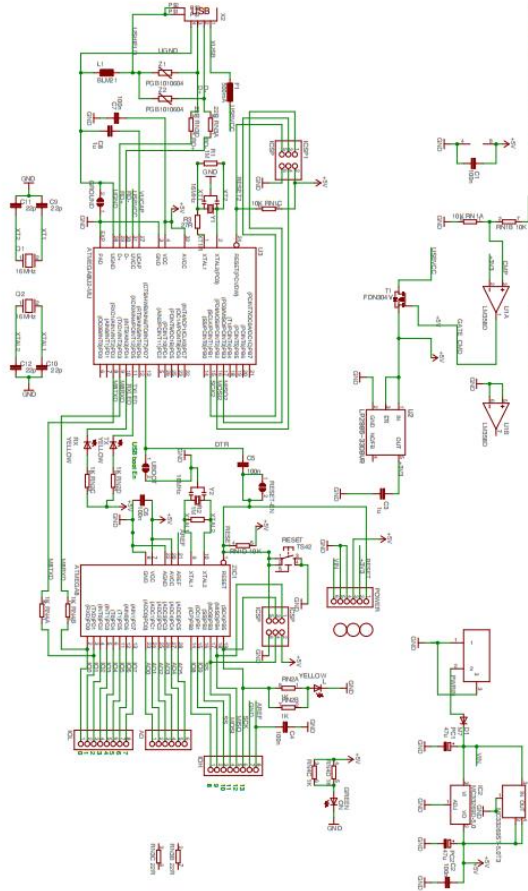
TX RX Flashing



Blinking Led!

Arduino™ UNO Reference Design

THIS DOCUMENT AND THE PROPOSED PCB ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THE USER ASSUMES ALL LIABILITY FOR ANY DAMAGE, INCLUDING BUT NOT LIMITED TO, ANY DAMAGE TO PERSONS OR PROPERTY, ARISING OUT OF OR FROM THE USE OF THE INFORMATION CONTAINED HEREIN, INCLUDING THE INFORMATION CONTAINED IN THIS DOCUMENT. THE USER ASSUMES ALL LIABILITY FOR ANY DAMAGE, INCLUDING BUT NOT LIMITED TO, ANY DAMAGE TO PERSONS OR PROPERTY, ARISING OUT OF OR FROM THE USE OF THE INFORMATION CONTAINED HEREIN, INCLUDING THE INFORMATION CONTAINED IN THIS DOCUMENT. THE USER ASSUMES ALL LIABILITY FOR ANY DAMAGE, INCLUDING BUT NOT LIMITED TO, ANY DAMAGE TO PERSONS OR PROPERTY, ARISING OUT OF OR FROM THE USE OF THE INFORMATION CONTAINED HEREIN, INCLUDING THE INFORMATION CONTAINED IN THIS DOCUMENT.



2. Datasheet AD8232

Data Sheet

AD8232

SPECIFICATIONS

 $V_S = 3\text{ V}$, $V_{REF} = 1.5\text{ V}$, $V_{CM} = 1.5\text{ V}$, $T_A = 25^\circ\text{C}$, $FB = \text{low}$, $SDN = \text{high}$, $AC/DC = \text{low}$, unless otherwise noted.

Table 1.

Parameter	Symbol	Test Conditions/Comments	Min	Typ	Max	Unit
INSTRUMENTATION AMPLIFIER						
Common-Mode Rejection Ratio, DC to 60 Hz	CMRR	$V_{CM} = 0.35\text{ V to }2.85\text{ V}$, $V_{REF} = 0\text{ V}$	80	86		dB
		$V_{CM} = 0.35\text{ V to }2.85\text{ V}$, $V_{REF} = \pm 0.3\text{ V}$		80		dB
Power Supply Rejection Ratio	PSRR	$V_S = 2.0\text{ V to }3.5\text{ V}$	76	90		dB
Offset Voltage (RTI)	V_{OS}			3	8	mV
Instrumentation Amplifier Inputs DC Blocking Input ¹				5	50	μV
Average Offset Drift				10		$\mu\text{V}/^\circ\text{C}$
Instrumentation Amplifier Inputs DC Blocking Input ²				0.05		$\mu\text{V}/^\circ\text{C}$
Input Bias Current	I_b			50	200	pA
		$T_A = 0^\circ\text{C to }70^\circ\text{C}$		1		nA
Input Offset Current	I_{OS}			25	100	pA
		$T_A = 0^\circ\text{C to }70^\circ\text{C}$		1		nA
Input Impedance				10 7.5		G pF
Differential				5 15		G pF
Common Mode						
Input Voltage Noise (RTI)		$f = 1\text{ kHz}$		100		nV/ $\sqrt{\text{Hz}}$
Spectral Noise Density		$f = 0.1\text{ Hz to }10\text{ Hz}$		12		$\mu\text{V p-p}$
Peak-to-Peak Voltage Noise		$f = 0.5\text{ Hz to }40\text{ Hz}$		14		$\mu\text{V p-p}$
Input Voltage Range		$T_A = 0^\circ\text{C to }70^\circ\text{C}$	0.2		$+V_S$	V
DC Differential Input Range	V_{REF}		-300		+300	mV
Output						
Output Swing		$R_L = 50\text{ k}\Omega$	0.1		$+V_S - 0.1$	V
Short-Circuit Current	I_{OUT}			6.3		mA
Gain	A_V			100		V/V
Gain Error		$V_{REF} = 0\text{ V}$		0.4		%
		$V_{REF} = -300\text{ mV to }+300\text{ mV}$		1	3.5	%
Average Gain Drift		$T_A = 0^\circ\text{C to }70^\circ\text{C}$		12		ppm/ $^\circ\text{C}$
Bandwidth	BW			2		kHz
RFI Filter Cutoff (Each Input)				1		MHz
OPERATIONAL AMPLIFIER (A1)						
Offset Voltage	V_{OS}			1	5	mV
Average TC		$T_A = 0^\circ\text{C to }70^\circ\text{C}$		5		$\mu\text{V}/^\circ\text{C}$
Input Bias Current	I_b			100		pA
		$T_A = 0^\circ\text{C to }70^\circ\text{C}$		1		nA
Input Offset Current	I_{OS}			100		pA
		$T_A = 0^\circ\text{C to }70^\circ\text{C}$		1		nA
Input Voltage Range			0.1		$+V_S - 0.1$	V
Common-Mode Rejection Ratio	CMRR	$V_{CM} = 0.5\text{ V to }2.5\text{ V}$		100		dB
Power Supply Rejection Ratio	PSRR			100		dB
Large Signal Voltage Gain	A_{VOL}			110		dB
Output Voltage Range		$R_L = 50\text{ k}\Omega$	0.1		$+V_S - 0.1$	V
Short-Circuit Current Limit	I_{OUT}			12		mA
Gain Bandwidth Product	GBP			100		kHz
Slew Rate	SR			0.02		V/ μs
Voltage Noise Density (RTI)	e_n	$f = 1\text{ kHz}$		60		nV/ $\sqrt{\text{Hz}}$
Peak-to-Peak Voltage Noise (RTI)	$e_{n,pp}$	$f = 0.1\text{ Hz to }10\text{ Hz}$		6		$\mu\text{V p-p}$
		$f = 0.5\text{ Hz to }40\text{ Hz}$		8		$\mu\text{V p-p}$

Rev. B | Page 3 of 20

Parameter	Symbol	Test Conditions/Comments	Min	Typ	Max	Unit
RIGHT LEG DRIVE AMPLIFIER (A2)						
Output Swing		$R_L = 50\text{ k}\Omega$	0.1		$+V_S - 0.1$	V
Short-Circuit Current	I_{out}			11		mA
Integrator Input Resistor			120	150	180	k Ω
Gain Bandwidth Product	GDP			100		kHz
REFERENCE BUFFER (A3)						
Offset Error	V_{os}	$R_L > 50\text{ k}\Omega$		1		mV
Input Bias Current	I_b			100		pA
Short-Circuit Current Limit	I_{out}			12		mA
Voltage Range		$R_L = 50\text{ k}\Omega$	0.1		$+V_S - 0.7$	V
DC LEADS OFF COMPARATORS						
Threshold Voltage				$+V_S - 0.5$		V
Hysteresis				60		mV
Propagation Delay				0.5		μ s
AC LEADS OFF DETECTOR						
Square Wave Frequency	F_{ac}		50	100	175	kHz
Square Wave Amplitude	I_{ac}			200		nA p-p
Impedance Threshold		Between +IN and -IN	10	20		M Ω
Detection Delay				110		μ s
FAST RESTORE CIRCUIT						
Switches		S1 and S2				
On Resistance	R_{on}		8	10	12	k Ω
Off Leakage				100		pA
Window Comparator						
Threshold Voltage		From either rail		50		mV
Propagation Delay				2		μ s
Switch Timing Characteristics						
Feedback Recovery Switch On Time	t_{on1}			110		ms
Filter Recovery Switch On Time	t_{on2}			55		ms
Fast Restore Reset	t_{nr}			2		μ s
LOGIC INTERFACE						
Input Characteristics						
Input Voltage (AC/DC and FR)						
Low	V_{IL}			1.24		V
High	V_{IH}			1.35		V
Input Voltage (SDN)						
Low	V_{IL}			2.1		V
High	V_{IH}			0.5		V
Output Characteristics		LOD+ and LOD- terminals				
Output Voltage						
Low	V_{OL}			0.05		V
High	V_{OH}			2.95		V
SYSTEM SPECIFICATIONS						
Quiescent Supply Current		$T_A = 0^\circ\text{C}$ to 70°C		170	230	μ A
Shutdown Current				210		μ A
		$T_A = 0^\circ\text{C}$ to 70°C		40	500	nA
Supply Range			2.0		3.5	V
Specified Temperature Range			0		70	$^\circ\text{C}$
Operational Temperature Range			-40		+85	$^\circ\text{C}$

¹ Offset referred to the input of the instrumentation amplifier inputs. See the Input Referred Offsets section for additional information.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

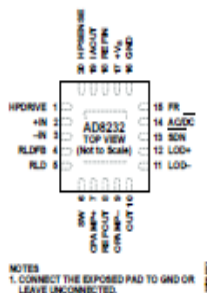


Figure 2. Pin Configuration

Table 3. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	HPDRIVE	High-Pass Driver Output. Connect HPDRIVE to the capacitor in the first high-pass filter. The AD8232 drives this pin to keep HPSENSE at the same level as the reference voltage.
2	+IN	Instrumentation Amplifier Positive Input. +IN is typically connected to the left arm (LA) electrode.
3	-IN	Instrumentation Amplifier Negative Input. -IN is typically connected to the right arm (RA) electrode.
4	RLDFB	Right Leg Drive Feedback Input. RLDFB is the feedback terminal for the right leg drive circuit.
5	RLD	Right Leg Drive Output. Connect the driven electrode (typically, right leg) to the RLD pin.
6	SW	Fast Restore Switch Terminal. Connect this terminal to the output of the second high-pass filter.
7	OPAMP+	Operational Amplifier Noninverting Input.
8	REFOUT	Reference Buffer Output. The instrumentation amplifier output is referenced to this potential. Use REFOUT as a virtual ground for any point in the circuit that needs a signal reference.
9	OPAMP-	Operational Amplifier Inverting Input.
10	OUT	Operational Amplifier Output. The fully conditioned heart rate signal is present at this output. OUT can be connected to the Input of an ADC.
11	LOD-	Leads Off Comparator Output. In dc leads off detection mode, LOD- is high when the electrode to -IN is disconnected, and it is low when connected. In ac leads off detection mode, LOD- is always low.
12	LOD+	Leads Off Comparator Output. In dc leads off detection mode, LOD+ is high when the +IN electrode is disconnected, and it is low when connected. In ac leads off detection mode, LOD+ is high when either the -IN or +IN electrode is disconnected, and it is low when both electrodes are connected.
13	SDN	Shutdown Control Input. Drive SDN low to enter the low power shutdown mode.
14	AC/DC	Leads Off Mode Control Input. Drive the AC/DC pin low for dc leads off mode. Drive the AC/DC pin high for ac leads off mode.
15	FR	Fast Restore Control Input. Drive FR high to enable fast recovery mode; otherwise, drive it low.
16	GND	Power Supply Ground.
17	+V	Power Supply Terminal.
18	REFIN	Reference Buffer Input. Use REFIN, a high impedance input terminal, to set the level of the reference buffer.
19	IAOUT	Instrumentation Amplifier Output Terminal.
20	HPSENSE	High-Pass Sense Input for Instrumentation Amplifier. Connect HPSENSE to the junction of R and C that sets the corner frequency of the dc blocking circuit.
	EP	Exposed Pad. Connect the exposed pad to GND or leave it unconnected.

3. Datasheet FT232R

3.1 Block Diagram (Simplified)

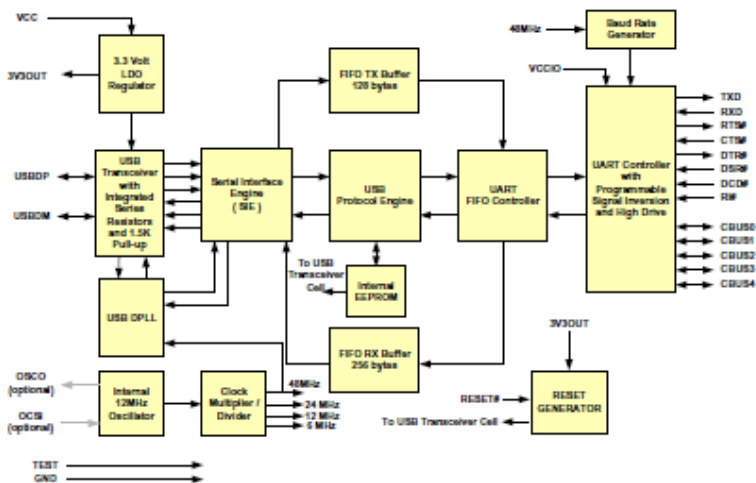


Figure 1 - FT232R Block Diagram

3.2 Functional Block Descriptions

3.3V LDO Regulator - The 3.3V LDO Regulator generates the 3.3V reference voltage for driving the USB transceiver cell output buffers. It requires an external decoupling capacitor to be attached to the 3V3OUT regulator output pin. It also provides 3.3V power to the 1.5KΩ internal pull up resistor on USBDP. The main function of this block is to power the USB Transceiver and the Reset Generator Cells rather than to power external logic. However, external circuitry requiring a 3.3V nominal supply at a current of around than 50mA could also draw its power from the 3V3OUT pin, if required.

USB Transceiver - The USB Transceiver Cell provides the USB 1.1 / USB 2.0 full-speed physical interface to the USB cable. The output drivers provide 3.3V level slew rate control signalling, whilst a differential receiver and two single ended receivers provide USB data in, SEO and USB Reset condition detection. This Cell also incorporates internal USB series resistors on the USB data lines, and a 1.5KΩ pull up resistor on USBDP.

USB DPLL - The USB DPLL cell locks on to the incoming NRZI USB data and provides separate recovered clock and data signals to the SIE block.

Internal 12MHz Oscillator - The Internal 12MHz Oscillator cell generates a 12MHz reference clock input to the x4 Clock multiplier. The 12MHz Oscillator is also used as the reference clock for the SIE, USB Protocol Engine and UART FIFO controller blocks.

Clock Multiplier / Divider - The Clock Multiplier / Divider takes the 12MHz input from the Oscillator Cell and generates the 48MHz, 24MHz, 12MHz, and 6MHz reference clock signals. The 48MHz clock reference is used for the USB DPLL and the Baud Rate Generator blocks.

Serial Interface Engine (SIE) - The Serial Interface Engine (SIE) block performs the Parallel to Serial and Serial to Parallel conversion of the USB data. In accordance to the USB 2.0 specification, it performs bit stuffing / un-stuffing and CRC5 / CRC16 generation / checking on the USB data stream.

USB Protocol Engine - The USB Protocol Engine manages the data stream from the device USB control endpoint. It handles the low level USB protocol (Chapter 9) requests generated by the USB host controller and the commands for controlling the functional parameters of the UART.

FIFO TX Buffer (128 bytes) - Data from the USB data out endpoint is stored in the FIFO TX buffer and removed from the buffer to the UART transmit register under control of the UART FIFO controller.

FIFO RX Buffer (256 bytes) - Data from the UART receive register is stored in the FIFO RX buffer prior to being removed by the SIE on a USB request for data from the device data in endpoint.

UART FIFO Controller - The UART FIFO controller handles the transfer of data between the FIFO RX and TX buffers and the UART transmit and receive registers.

UART Controller with Programmable Signal Inversion and High Drive - Together with the UART FIFO Controller the UART Controller handles the transfer of data between the FIFO RX and FIFO TX buffers and the UART transmit and receive registers. It performs asynchronous 7 / 8 bit Parallel to Serial and Serial to Parallel conversion of the data on the RS232 (RS422 and RS485) Interface. Control signals supported by UART mode include RTS, CTS, DSR, DTR, DCD and RI. The UART Controller also provides a transmitter enable control signal pin option (TXDEN) to assist with interfacing to RS485 transceivers. RTS / CTS, DSR / DTR and X-On / X-Off handshaking options are also supported. Handshaking, where required, is handled in hardware to ensure fast response times. The UART also supports the RS232 BREAK setting and detection conditions. A new feature, programmable in the Internal EEPROM allows the UART signals to each be individually inverted. Another new EEPROM programmable feature allows a high signal drive strength to be enabled on the UART Interface and CBUS pins.

Baud Rate Generator - The Baud Rate Generator provides a x16 clock input to the UART Controller from the 48MHz reference clock and consists of a 14 bit prescaler and 3 register bits which provide fine tuning of the baud rate (used to divide by a number plus a fraction or "sub-integer"). This determines the Baud Rate of the UART, which is programmable from 183 baud to 3 million baud.

The FT232R supports all standard baud rates and non-standard baud rates from 300 Baud up to 3 Megabaud. Achievable non-standard baud rates are calculated as follows -

$$\text{Baud Rate} = 3000000 / (n + x)$$

where n can be any integer between 2 and 16,384 ($= 2^{14}$) and x can be a sub-integer of the value 0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, or 0.875. When $n = 1$, $x = 0$, i.e. baud rate divisors with values between 1 and 2 are not possible.

This gives achievable baud rates in the range 183.1 baud to 3,000,000 baud. When a non-standard baud rate is required simply pass the required baud rate value to the driver as normal, and the FTDI driver will calculate the required divisor, and set the baud rate. See FTDI application note AN232B-05 for more details.

RESET Generator - The Integrated Reset Generator Cell provides a reliable power-on reset to the device internal circuitry on power up. A RESET# input pin is provided to allow other devices to reset the FT232R. RESET# can be tied to VCCIO or left unconnected, unless it is a requirement to reset the device from external logic or an external reset generator I.C.

Internal EEPROM - The Internal EEPROM in the FT232R can be used to store USB Vendor ID (VID), Product ID (PID), device serial number, product description string, and various other USB configuration descriptors. The Internal EEPROM is also used to configure the CBUS pin functions. The device is supplied with the Internal EEPROM settings preprogrammed as described in [Section 10](#).

4. Device Pin Out and Signal Descriptions

4.1 28-LD SSOP Package

Page 7

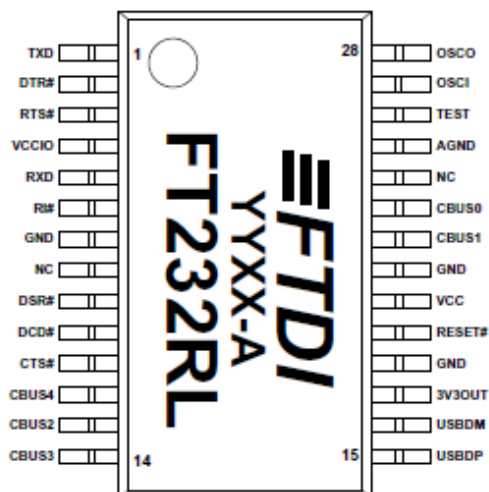
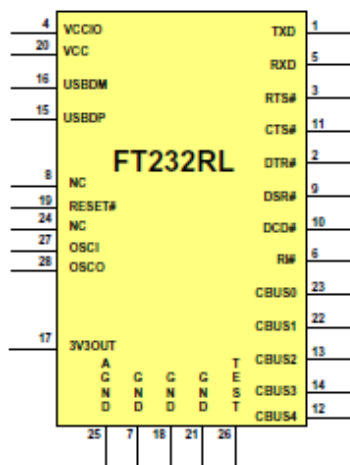


Figure 2 - 28 Pin SSOP Package Pin Out



4.2 SSOP-28 Package Signal Descriptions

Table 1 - SSOP Package Pin Out Description

Pin No.	Name	Type	Description
USB Interface Group			
15	USBDP	I/O	USB Data Signal Plus, incorporating internal series resistor and 1.5kΩ pull up resistor to 3.3V
16	USBDM	I/O	USB Data Signal Minus, incorporating internal series resistor.
Power and Ground Group			
4	VCCIO	PWR	+1.8V to +5.25V supply to the UART interface and CBUS group pins (1...3, 5, 6, 9...14, 22, 23). In USB bus-powered designs connect to 3V3OUT to drive out at 3.3V levels, or connect to VCC to drive out at 5V CMOS level. This pin can also be supplied with an external 1.8V - 2.8V supply in order to drive out at lower levels. It should be noted that in this case this supply should originate from the same source as the supply to Vcc. This means that in bus-powered designs a regulator which is supplied by the 5V on the USB bus should be used.
7, 16, 21	GND	PWR	Device ground supply pins
17	3V3OUT	Output	3.3V output from integrated L.D.O. regulator. This pin should be decoupled to ground using a 100nF capacitor. The prime purpose of this pin is to provide the internal 3.3V supply to the USB transceiver cell and the internal 1.5kΩ pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the FT232RL's VCCIO pin.
20	VCC	PWR	3.3V to 5.25V supply to the device core.
25	AGND	PWR	Device analog ground supply for internal clock multiplier
Miscellaneous Signal Group			
8, 24	NC	NC	No internal connection.
19	RESET#	Input	Can be used by an external device to reset the FT232RL. If not required can be left unconnected, or pulled up to VCCIO.
26	TEST	Input	Put the device into I.C. test mode. Must be tied to GND for normal operation.
27	OSCI	Input	Input to 12MHz Oscillator Cell. Optional - Can be left unconnected for normal operation. *
28	OSCO	Output	Output from 12MHz Oscillator Cell. Optional - Can be left unconnected for normal operation if internal oscillator is used. *
UART Interface and CBUS Group **			
1	TXD	Output	Transmit Asynchronous Data Output.
2	DTR#	Output	Data Terminal Ready Control Output / Handshake signal.
3	RTS#	Output	Request To Send Control Output / Handshake signal.
5	RXD	Input	Receive Asynchronous Data Input.
6	RI#	Input	Ring Indicator Control Input. When remote wake up is enabled in the internal EEPROM taking RI# low can be used to resume the PC USB host controller from suspend.
9	DSR#	Input	Data Set Ready Control Input / Handshake signal.
10	DCD#	Input	Data Carrier Detect Control Input.
11	CTS#	Input	Clear to Send Control Input / Handshake signal.
12	CBUS4	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is SLEEP#. See CBUS Signal Options, Table 3.
13	CBUS2	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is TXDEN. See CBUS Signal Options, Table 3.
14	CBUS3	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is PWREN#. See CBUS Signal Options, Table 3.
22	CBUS1	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is RXLED#. See CBUS Signal Options, Table 3.
23	CBUS0	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is TXLED#. See CBUS Signal Options, Table 3.

* Contact [FTDI technical support](#) for details on how to use an external crystal, ceramic resonator, or oscillator with the FT232RL.

** When used in Input Mode, these pins are pulled to VCCIO via internal 200kΩ resistors. These pins can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting an option in the internal EEPROM.

8.4 USB to MCU UART Interface

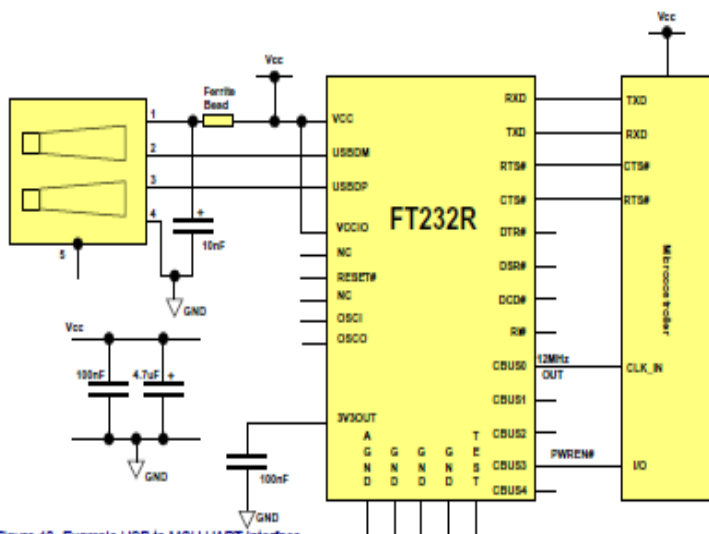


Figure 18 -Example USB to MCU UART Interface

Figure 18 is an example of interfacing the FT232R to a Microcontroller (MCU) UART Interface. This example uses TXD and RXD for transmission and reception of data, and RTS# / CTS# hardware handshaking. Also in this example CBUS0 has been configured as a 12MHz output which is being used to clock the MCU.

Optionally, Ri# can be connected to another I/O pin on the MCU and could be used to wake up the USB host controller from suspend mode. If the MCU is handling power management functions, then a CBUS pin can be configured as PWREN# and should also be connected to an I/O pin of the MCU.

LAMPIRAN B PROGRAM

1. Program pada *Software* IDE

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(3, 4); // RX, TX

int data,sat,pul,rat,rib;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(19200);
  while (!Serial)
  {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // set the data rate for the SoftwareSerial port
  mySerial.begin(19200);
  pinMode(10, INPUT); // Setup for leads off detection LO +
  pinMode(11, INPUT); // Setup for leads off detection LO -
}

void loop() { // run over and over

  if (mySerial.available())
  {
    Serial.write(mySerial.read());
  }
  if (Serial.available())
  {
    mySerial.write(Serial.read());
  }
}
```

```

if((digitalRead(10) == 1)||digitalRead(11) == 1)){
  Serial.println('!');
}
else
{

  data = analogRead(A0);
  sat = data % 10;
  pul = (data/10) % 10;
  rat = (data/100) % 10;
  rib = data/1000;
  mySerial.print('a');
  mySerial.print(rib);
  mySerial.print(rat);
  mySerial.print(pul);
  mySerial.println(sat);

}
delayMicroseconds(1);
}

```

3. Program QRS *Detector* pada Delphi

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, CPort, Buttons, TeEngine, Series, ExtCtrls,
  TeeProcs,
  Chart, math;
type
  TForm1 = class(TForm)
    Chart1: TChart;
    Series1: TLineSeries;
    QRS: TSpeedButton;

```

```

BitBtn1: TBitBtn;
ComPort1: TComPort;
Edit1: TEdit;
Label1: TLabel;
Edit4: TEdit;
Label4: TLabel;
SpeedButton1: TSpeedButton;
Chart2: TChart;
Chart3: TChart;
Chart4: TChart;
Series5: TFastLineSeries;
Chart5: TChart;
Series6: TFastLineSeries;
Chart6: TChart;
Chart7: TChart;
Series7: TFastLineSeries;
Chart8: TChart;
Series8: TFastLineSeries;
SpeedButton2: TSpeedButton;
Edit2: TEdit;
Series4: TFastLineSeries;
Series3: TFastLineSeries;
procedure QRSClick(Sender: TObject);
procedure delay(lama:longint);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure EKG;procedure LPF; procedure HPF;
Procedure Derivative;procedure squaring;
procedure Moving;

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  sumbuX,sumbuY,sumbuZ : array[0..400] of integer;

```



```

ecg : array[0..200] of integer;
y1,y2,y3,y4,y5 : array[0..200] of real;
i,t,m: integer;
temp : real;
VX : real;
sudutX :real;
implementation

{$R *.dfm}
procedure TForm1.delay(lama:longint);
var ref:longint;
begin
    ref:=gettickcount;
    repeat application.processmessages;
    until ((gettickcount-ref)>=lama);
end;
procedure TForm1.QRSClick(Sender: TObject);
var
    cek,rib1,rat1,pul1,sat1,rib2,rat2,pul2,sat2 : string;
    rib,rat,pul,sat : integer;
    V1,V2,V3: real;

label
cetak,hapus;
begin
    if QRS.Caption = 'Stop' then
        QRS.Caption :='Run'
    else
        if QRS.Caption = 'Run' then
            begin
                QRS.Caption :='Stop';
                exit;
            end;
        Comport1.Open;i:=0;
        Series1.Clear;

        repeat
            repeat

```

```

    Comport1.ReadStr(cek,1);
    delay(1);
until cek = 'a';

Comport1.ReadStr(rib1,4);
//Comport1.ReadStr(rib2,4);

rib := StrToIntDef(rib1,0);
sumbuX[i] := rib;
V1:= rib * (5/1023);
Edit4.Text := FloatToStr(V1);
Edit1.Text := IntToStr(sumbuX[i]);

i:=i+1;

if i=400 then
begin
Series1.Clear;

for i:=0 to 400 do
begin
Series1.AddXY(i,sumbuX[i],"clRed);

Edit2.Text :=IntToStr(i) ;
delay(10);
end;

hapus:
i:=0;
delay(10);

end;
until QRS.Caption ='Stop';
Comport1.Close;

```

```

    end;
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    Comport1.ShowSetupDialog;
end;

procedure TForm1.EKG;
begin
    for t:=1 to 200 do
    begin
        ecg[t]:=0;
    end;
    for t:=1 to 200 do
    begin
        ecg[t]:=sumbuX[t];
    end;
end;

end;

procedure TForm1.LPF;
begin
    for t:= 1 to 200 do
    begin
        if t = 1 then y1[t]:= ecg[t]
        else
            if t = 2 then y1[t]:= 2*y1[t-1]+ ecg[t]
            else
                if t<= 6 then y1[t]:= 2*y1[t-1]-y1[t-2]+ ecg[t]
                else
                    if t<= 12 then y1[t]:=2*y1[t-1]-y1[t-2]+ ecg[t]-2*ecg[t-6]
                    else
                        y1[t]:=2*y1[t-1]-y1[t-2]+ ecg[t]-2*ecg[t-6]+ ecg[t-12];
                    end;
            end;
        for t:=0 to 200 do
        begin
            Series4.Addxy(t,y1[t],"clred");
            delay(1);
        end;
    end;
end;

```

```
end;  
end;
```

```
procedure TForm1.HPF;  
begin  
  for t:= 1 to 200 do  
    begin  
      if t = 1 then y2[t] :=-y1[t]/32  
      else  
        if t<=16 then y2[t] :=-(y2[t-1]+y1[t])/32  
        else  
          if t<=32 then y2[t] := y1[t-16]-(y2[t-1]+y1[t])/32  
          else  
            y2[t]:=y1[t-16]-(y2[t-1]+y1[t]-y1[t-32])/32;  
          end;  
        end;  
      for t:=0 to 200 do  
        begin  
          Series5.Addxy(t,y2[t],"clred");  
          delay(1);  
        end;  
      end;  
    end;  
  end;
```

```
Procedure TForm1.Derivative;  
begin  
  for t :=1 to 200 do  
    begin  
      if t=1 then y3[t]:=2*y2[t]/8  
      else  
        if t<=3 then y3[t]:=(2*y2[t]+y2[t-1])/8  
        else  
          if t<=4 then y3[t]:=(2*y2[t]+y2[t-1]-y2[t-3])/8  
          else  
            y3[t]:=(2*y2[t]+ y2[t-1]-y2[t-3]-2*y2[t-4])/8;  
          end;  
        end;  
      for t:=0 to 200 do  
        begin  
          Series6.Addxy(t,y3[t],"clred");  
          delay(1);  
        end;  
      end;  
    end;  
  end;
```

```

    end;
end;

procedure TForm1.Squaring;
begin
    for t:=1 to 200 do
        y4[t]:=y3[t] * y3[t];
    for t:=0 to 200 do
        begin
            Series7.Addxy(t,y4[t],"clred");
            delay(1);
        end;
    end;
procedure TForm1.Moving;
begin
    for t :=1 to 200 do
        begin
            if t =1 then y5[t] :=y4[t]/16
            else
                if t <=8 then
                    begin
                        temp:=0;
                        for m :=1 to t-1 do
                            begin
                                temp:=temp+y4[t-m];
                            end;
                        y5[t]:=(y4[t]+temp)/16;
                    end
                else
                    temp:=0;
                    for m:=1 to 7 do
                        begin
                            temp:=temp+y4[t-m];
                        end;
                        y5[t]:=(y4[t]+temp)/16;
                    end;
                end;
            for t:=0 to 200 do
                begin
                    Series8.Addxy(t,y5[t],"clred");

```

```

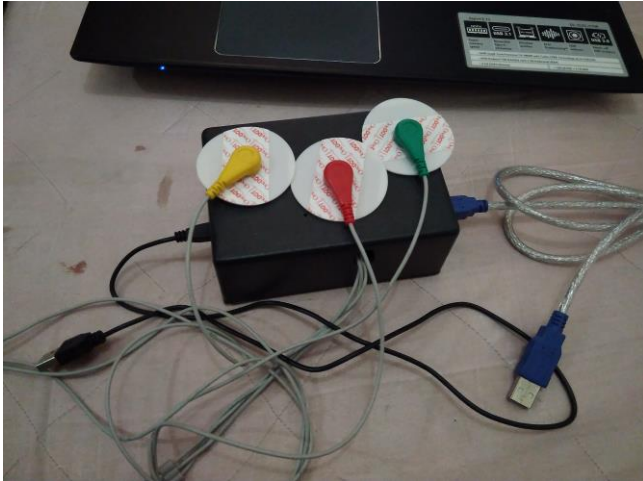
    delay(1);
end;
end;
procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    Series3.clear;Series4.clear;Series5.clear;
    Series6.clear;Series7.clear;Series8.clear;
    EKG;
    for i:=1 to 200 do
    begin
        Series3.Addxy(i,ecg[i],",clred");
        delay(1);
    end;
    LPF;
    HPF;
    Derivative;
    Squaring;
    Moving;
end;

end.

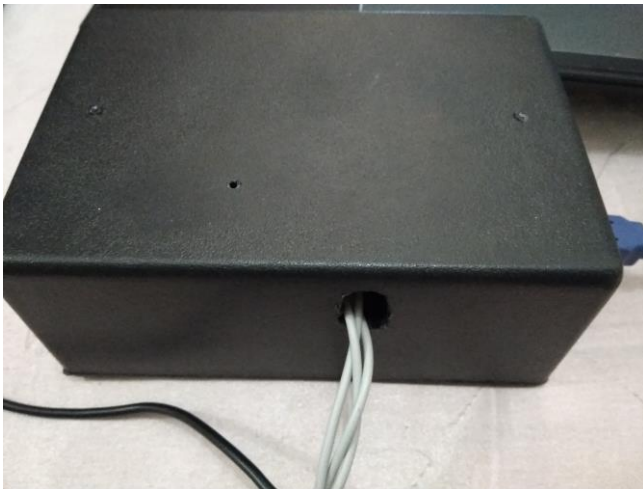
```

LAMPIRAN C FOTO

1. Instrument Detektor Arithmia



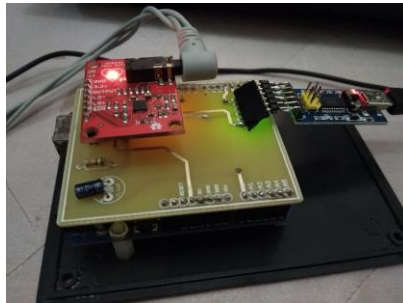
2. Box Detektor Arithmia



3. *Lead ECG*



4. Bagian Elektronik Instrument Detektor Arithmia



5. Proses Pengambilan Data



DAFTAR RIWAYAT HIDUP



Nama : Abdurrachman Hakim
TTL : Surabaya, 6 Oktober 1996
Jenis Kelamin : Laki-Laki
Agama : Islam
Alamat : Kupang Praupang Pasar
1/19, Surabaya
Telp/HP : 085731471971
E-mail : hakimabdurachman1@gmail.com

RIWAYAT PENDIDIKAN

1. 2002 – 2008 : SD Praja Mukti, Surabaya
2. 2008 – 2010 : MTs Akselerasi Amanatul Ummah, Pacet
3. 2010 – 2013 : MBI Amanatul Ummah, Pacet
4. 2013 – 2017 : Program Studi Teknik Elektronik Industri, Departemen Teknik Elektro Otomasi – Fakultas Vokasi, Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PTPN X Pabrik Gula Lestari, Kertosono, Nganjuk

PENGALAMAN ORGANISASI

1. Staff Hubungan Luar HimaD3Tekro 2014 – 2015
2. Kepala Biro Manajemen Pelatihan UKM Cinta Rebana ITS 2014 – 2015
3. Kepala Departemen PSDM UKM Cinta Rebana ITS 2015 – 2016
4. Dewan Penasihat dan Pertimbangan UKM Cinta Rebana ITS 2016 – 2017